

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

(NASA-CR-143848) AIRCRAFT MODEL PROTOTYPES
WHICH HAVE SPECIFIED HANDLING-QUALITY TIME
HISTORIES Final Report (Lehigh Univ.) 51 p
HC A04/MF 301 CSCL 31C

N78-14054

Unclass

G3/03 53656

NASA CR-143848

AIRCRAFT MODEL PROTOTYPES WHICH HAVE SPECIFIED
HANDLING-QUALITY TIME HISTORIES

S. H. Johnson

Lehigh University
Bethlehem, Pennsylvania 18015

January 1978

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Dryden Flight Research Center
Edwards, California 93523

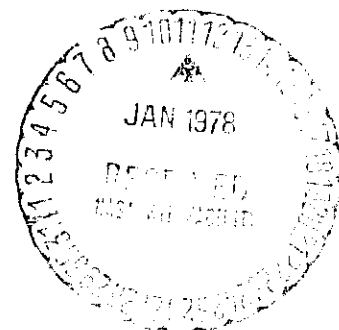


TABLE OF CONTENTS

List of Figures	v
List of Tables	vi
Symbols	vii
INTRODUCTION	1
PROBLEM STATEMENT	2
METHODS	4
COMPUTATIONAL PROCEDURES	22
ZERO SUPPRESSION	34
VERIFICATION	43
DISCUSSION	51
APPENDICES	
A. Use of Program AANDB	53
B. Program AANDB Listing	69
C. Program AANDB Sample Output	104
D. Program AANDB Sample Input	126
REFERENCES	143
CONVERSION FACTORS TO SI UNITS	144

LIST OF FIGURES

Figure

- | | |
|-------|---|
| 1 | C_n^* longitudinal handling-quality criterion envelopes |
| 2 | p_n normalized roll-rate envelopes |
| 3 | β_n normalized sideslip envelopes |
| 4A,B | D_n^* lateral handling-quality criterion envelopes |
| 5 | acceptable eigenvalues for second-order longitudinal models |
| 6 | fitted roll-rate and roll-acceleration responses |
| 7 | fitted sideslip and sideslip rate-of-change responses |
| 8A,B | fitted D_n^* and $\dot{D}_n^*$ rate-of-change responses |
| 9 | roll-rate pseudodata |
| 10 | roll-rate response of fourth-order model |
| 11 | sideslip response of fourth-order model |
| 12A,B | D_n^* response of fourth-order model |
| 13 | fitted roll rate with one zero suppressed |
| 14 | fitted roll rate with two zeros suppressed |
| 15 | fitted sideslip with one zero suppressed |
| 16 | fitted sideslip with two zeros suppressed |
| 17 | model roll rate and roll acceleration |
| 18 | model sideslip and sideslip rate-of-change |
| 19A,B | model D_n^* and $\dot{D}_n^*$ rate-of-change |
| A1 | program AANDB structure |
| A2 | model roll rate and roll acceleration |
| A3 | model sideslip and sideslip rate-of-change |
| A4A,B | model D_n^* and $\dot{D}_n^*$ rate-of-change |

LIST OF TABLES

Table

- | | |
|---|--|
| 1 | Unit correction factors for D^* equation |
| 2 | Jetstar parameters and flight conditions |
| 3 | Jetstar discrete time-history data |

SYMBOLS

$\underline{a}$	vector containing all of the elements of $\underline{A}$
$\underline{A}$	plant matrix with elements a_{ij}
$\underline{b}$	input distribution vector with elements b_i
$\underline{C}$	coefficients matrix with elements c_{ij}
$\underline{c}$	final-value vector with elements c_i
C_n^*	normalized longitudinal handling-quality criterion
$C^*(t)$	longitudinal handling-quality criterion
$D(\)$	denominator polynomial
D_n^*	normalized lateral handling-quality criterion
$D^*(t)$	lateral handling-quality criterion
d	differential operator $d \equiv \frac{d(\)}{dt}$
$\underline{G}$	output measurement matrix, matrix transfer function
$\tilde{\underline{G}}$	augmented output measurement vector matrix
$\underline{h}$	input/output coupling vector
$\tilde{\underline{h}}$	augmented input/output coupling vector
l	pilot station to vehicle C.G. distance
$\underline{N}(\)$	numerator polynomials vector
p_n	normalized roll-rate handling-quality time history
$p(t)$	roll rate
q_{co}	cross-over dynamic pressure in D^* definition
$r(t)$	yaw rate
s	Laplace transformation variable
$\underline{T}$	similarity transformation matrix
$u(t)$	forcing function

V nominal airspeed
 $\underline{x}$ state vector with elements $x_i(t)$
 $\underline{y}$ output vector with elements $y_i(t)$
 $\hat{\underline{y}}$ specified response vector with elements $\hat{y}_i(t)$
 $\tilde{\underline{y}}$ augmented output vector
 β_n normalized sideslip handling-quality time history
 $\beta(t)$ sideslip
 δ_a aileron deflection
 δ_e elevator deflection
 $\underline{A}$ eigenmatrix
 $\hat{\underline{A}}$ specified eigenmatrix
 $\underline{\lambda}$ eigenvalue vector with elements λ_j
 $\phi(t)$ roll angle

INTRODUCTION

Aircraft control systems have been designed in the past to meet stability, apparent damping, and sensitivity criteria. Subsequently, pilot comments and ratings, using the Cooper-Harper rating scale, graded the performance of the pilot-airplane-flight-control-system combination. The C-H scale emphasizes

TAKEOFF

CRUISE

GROSS MANEUVERING AND AEROBATICS

FORMATION FLIGHT

TRACKING

GROUND-CONTROLLED APPROACH

LANDING APPROACH [1]*

Experience has shown that the design criteria are often a subset of the handling-quality criteria. In order to design flight-control systems which achieve low (desirable) C-H ratings, some explicit consideration of handling qualities must be incorporated into the design process. The work begun by the Boeing Airplane Company and extended by the McDonnell-Douglas Company has resulted in quantitative specifications which are intended to insure "good" aircraft handling qualities. These specifications take the form of envelopes within which selected time-histories must fall, e.g., normalized sideslip, normalized roll rate, and two blended quantities, C^* and D^* , containing the acceleration cues at the pilot station.

An evolutionary process has provided the basic design of most aircraft control systems. It is often the case that the selection of numerical values

*

The numbers in square brackets refer to bibliography entries.

ORIGINAL PAGE IS
OF POOR QUALITY

for the many parameters in a control system is more difficult than the establishment of the control system structure and modes of operation. To overcome this "tuning" problem, a model-matching optimization technique can be implemented on a large digital computer. Such a program adjusts specified parameters of a simulated airplane-flight-control-system combination so as to minimize some measure of the dynamic differences between the closed-loop airplane and a low-order model of a desirable prototype airplane. As a participant in the ASEE-NASA Summer Faculty Fellowship Program in the summers of 1974 and 1975, the principal investigator, working in the Vehicle Dynamics and Control Division of the NASA Flight Research Center, Edwards, California, developed a method of translating handling-quality time-history specifications into prototype aircraft models. These models are suitable for subsequent use with an FRC model-matching program: CONOPT.

The connection between specified time histories which fall within the established envelopes and numerical values for adjustable parameters onboard the aircraft is, in principle, established. The method is explained in the following sections and the lateral-motion case demonstrated.

PROBLEM STATEMENT

The NASA Flight Research Center computer program CONOPT, which resulted from the addition of the MIT Model Performance Index Design Program OPT to the in-house control system analysis program CONTROL, is a model-matching package. The user specifies the model in transfer-function form. The Model PI technique [2] requires that the model transfer function, if it has any zeros, should have the same number of excess poles over zeros as the closed-loop

airplane control system. If it has no zeros then the number of poles of the model transfer function should be equal to or less than the number of excess poles over zeros of the closed-loop airplane. Another condition on the model transfer function is that it must have reasonable eigenvalues. Thus the model is a linear, constant-coefficient ordinary differential equation in operator notation,

$$D(d)x(t) = N(d)u(t) \quad , \quad (1)$$

with loose bounds on the roots of,

$$D(\lambda_i) = 0 \quad , \quad (2)$$

and constraints on the coefficients of $N(d)$. In order to obtain the entire matrix transfer function in one operation the model will be represented in state variable form,

$$d\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}u \quad . \quad (3)$$

The problem is to determine an $\mathbf{A}$ and $\mathbf{b}$ combination which satisfies the loose bounds and constraints above and which describes a low-order prototype airplane model with "good" handling qualities. FRC program CONTROL can be used to obtain transfer functions from $\mathbf{A}$ and $\mathbf{b}$ for use by CONOPT.

The handling-quality time histories, $y_i(t)$, are linear combinations of state variables,

$$\mathbf{y}(t) = \mathbf{G}\mathbf{x}(t) \quad . \quad (4)$$

If all the $y_i(t)$ fall within their handling-quality time-history envelopes the model is appropriate for use with CONOPT.

The rigid-body equations of motion for symmetrical airplanes slightly disturbed from straight and level flight can be separated into two sets of four simultaneous first-order linear constant-coefficient ordinary differential equations [3]. One set describes the longitudinal motion, or motion in the plane of symmetry of a normally configured airplane. The other set describes the lateral motion. In each case,

$$d\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}u \quad , \quad (5)$$

where $\mathbf{A}$ is 4×4 and $\mathbf{b}$ is 4×1 if there is only one input.

Only one longitudinal handling-quality criterion has been established: C_n^* . Three lateral handling-quality criteria have been proposed: D_n^* , normalized roll rate, and normalized sideslip. An envelope for the first derivative of each handling-quality time history has also been proposed [4]. These eight envelopes are plotted in Figures 1 through 4. These handling-quality time-history envelopes imply standard inputs. The C^* response results from a step change in elevator position, δ_e , and the lateral handling-quality time responses result from a step change in aileron position, δ_a .

METHODS

An obvious procedure for obtaining models with handling-quality time histories which fall within specified envelopes is to:

1. Guess a model in terms of $\mathbf{A}$ and $\mathbf{b}$
2. Calculate $\mathbf{x}(t)$
3. Form $\mathbf{y}(t)$
4. Compare $\mathbf{y}(t)$ and $\dot{\mathbf{y}}(t)$ to the appropriate envelopes

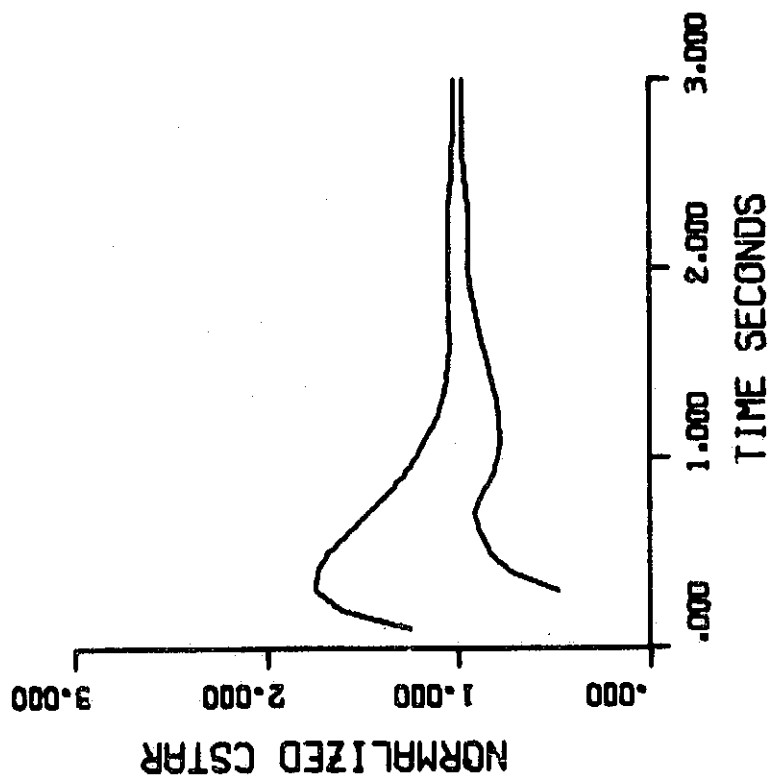
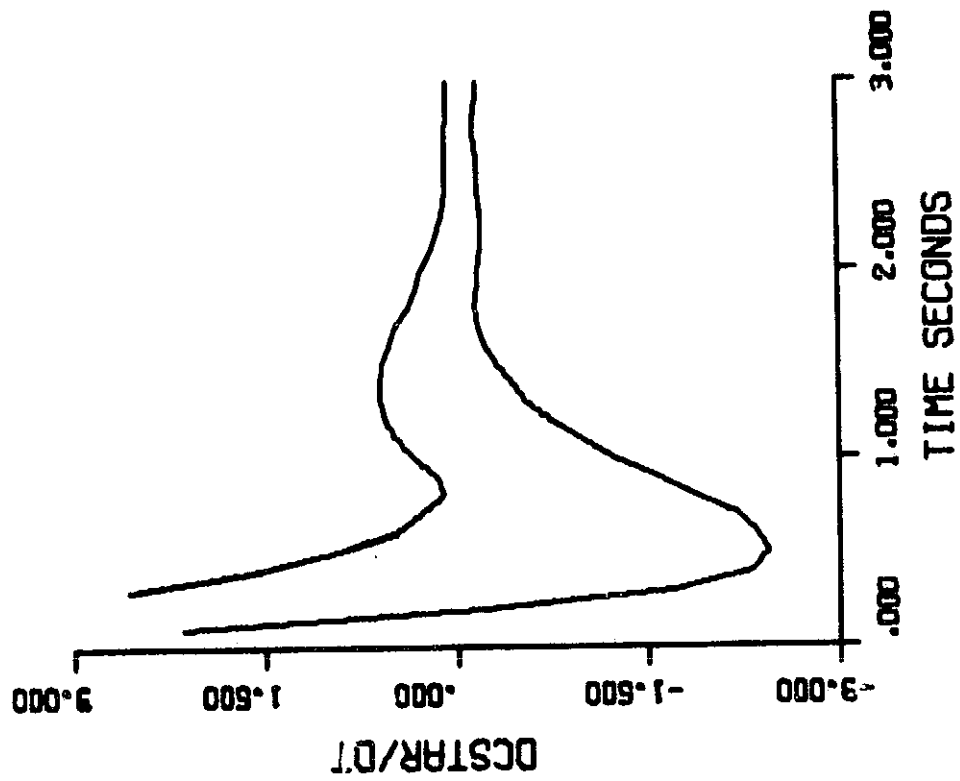


FIG.1 LONG. CRIT. ENVELOPES

ORIGINAL PAGE IS
OF POOR QUALITY

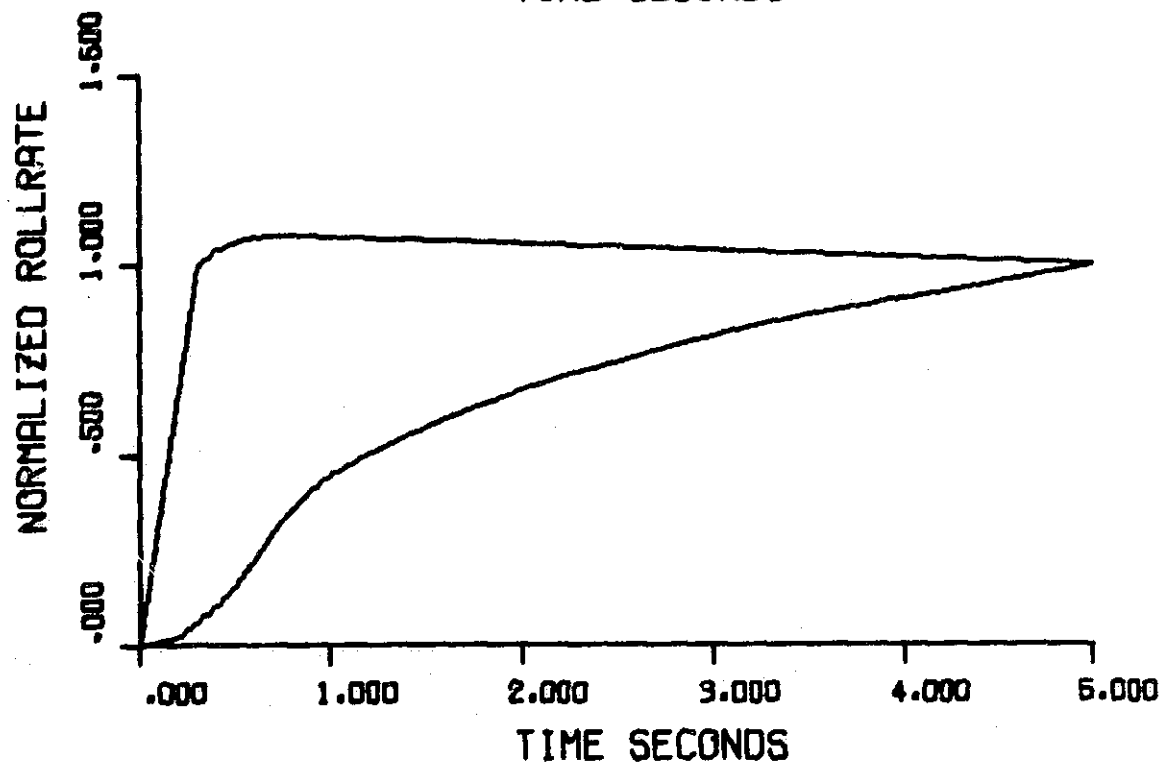
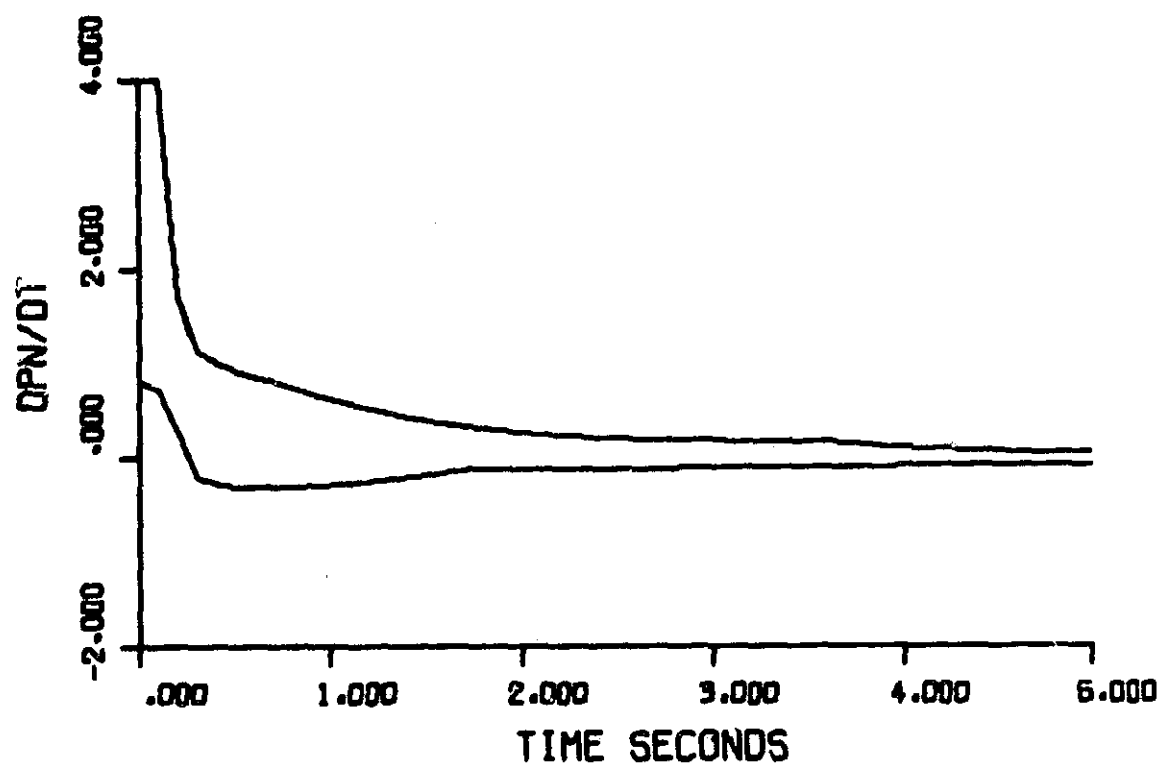


FIG.2 ROLLRATE ENVELOPES

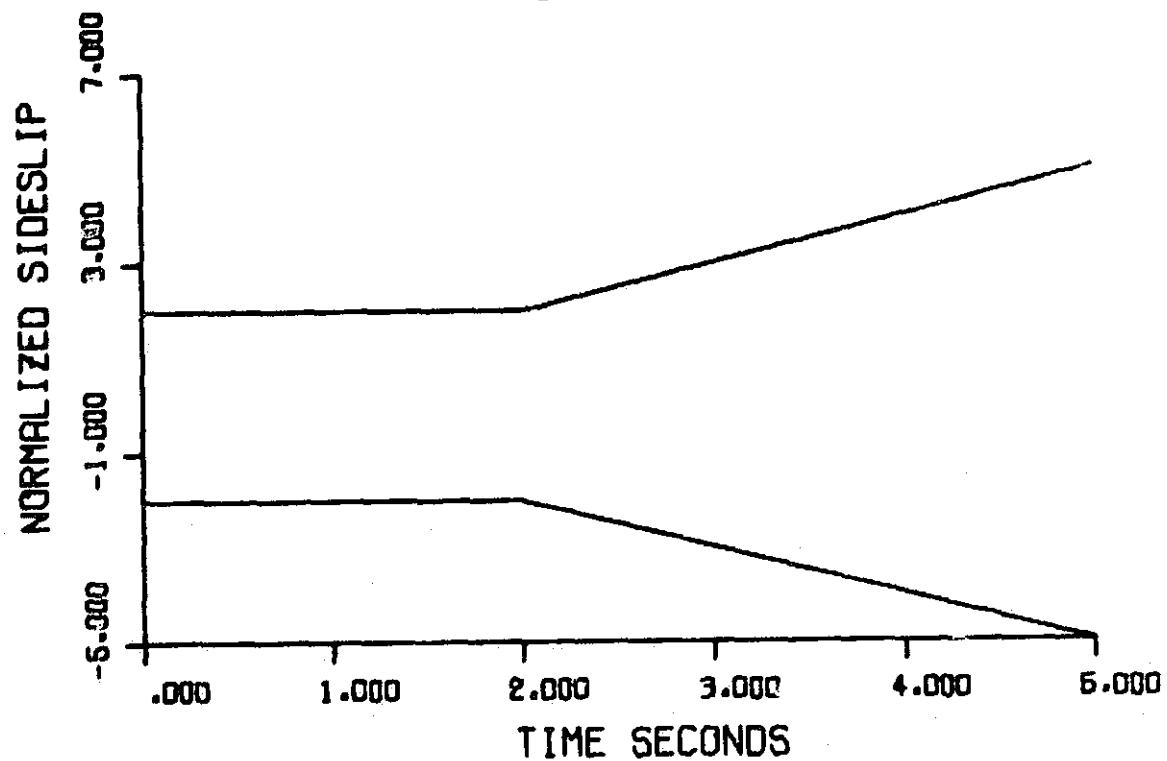
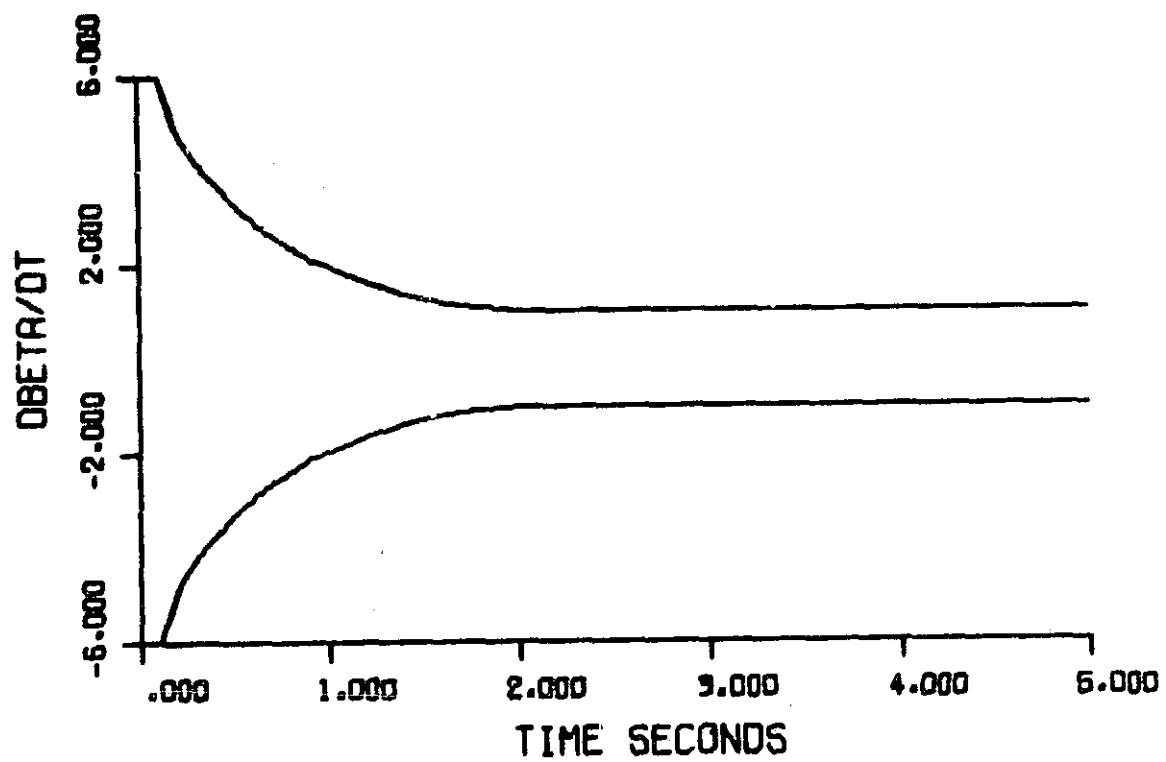


FIG.3 SIDESLIP ENVELOPES

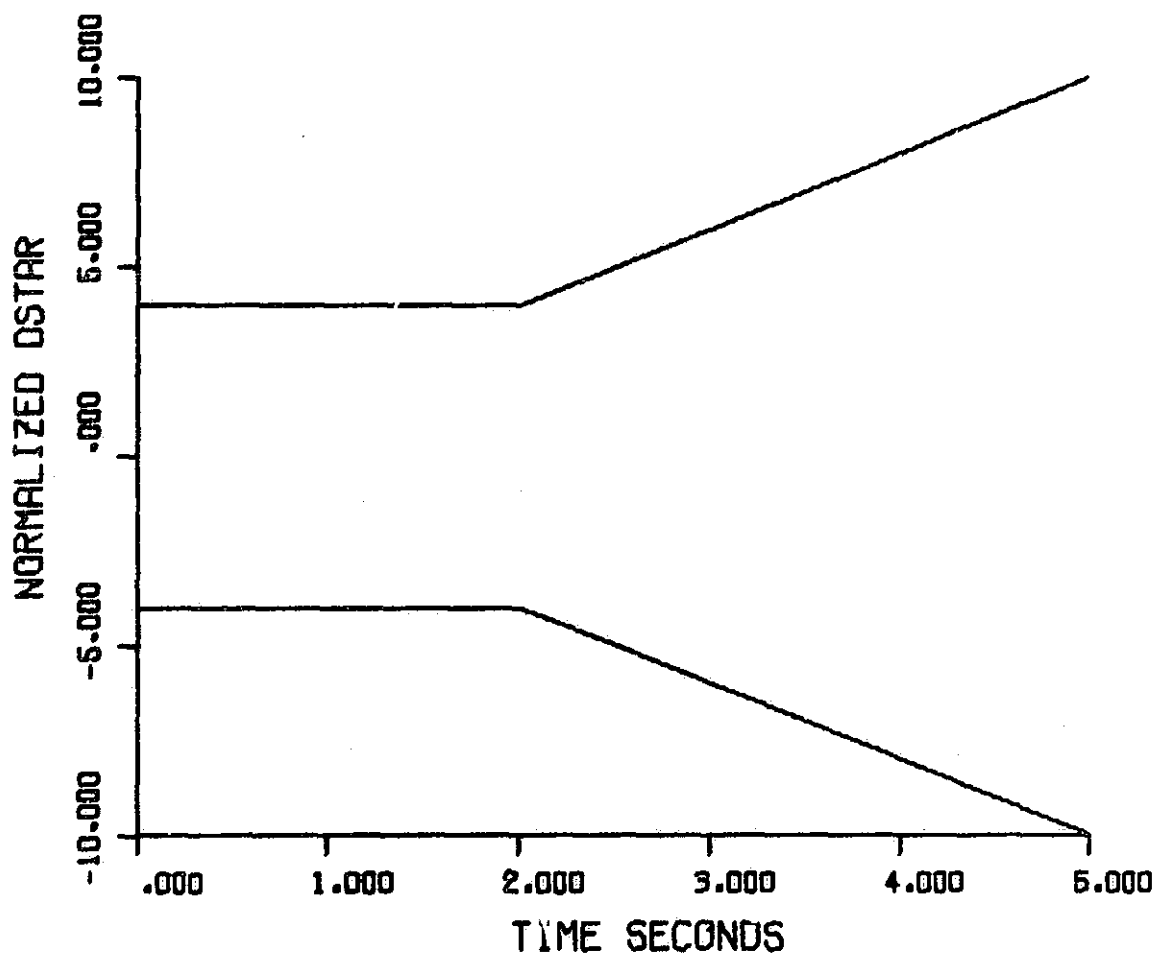


FIG.4A LATL. CRIT. ENVELOPES

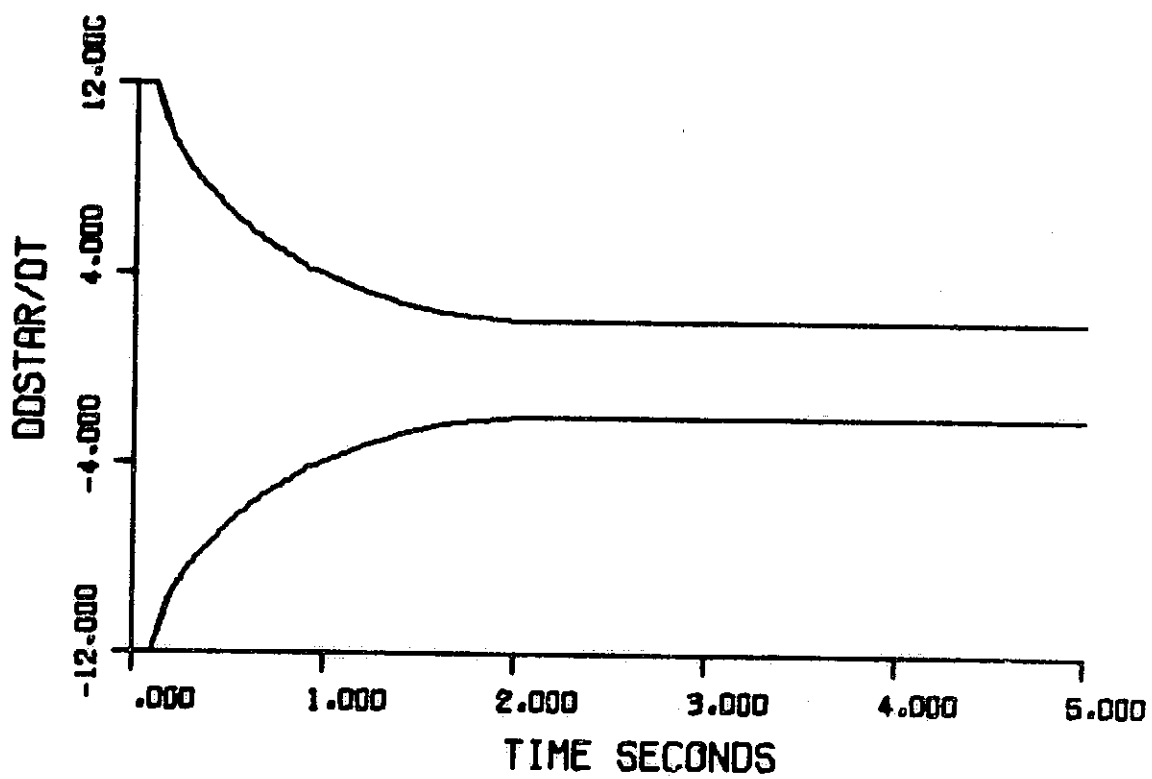


FIG. 4B

ORIGINAL PAGE IS
OF POOR QUALITY

5. If no envelopes are violated, stop
6. Otherwise alter A and/or b according to some strategy and return to Step 2.

This "brute force" technique would consume large amounts of computer time and may not succeed. It would be necessary to define a cost functional which measures all excursions outside of the envelopes and which is suitable for use in a numerical optimization scheme. Next a connection between the elements a_{ij} and b_i and the cost functional would have to be established. Finally, there is no assurance that the attainable minima of this functional would be zero. The addition of hard and soft constraints arising from the loose bounds on the eigenvalues and specification of the pole-zero excess of the transfer functions complicates the numerical optimization problem. In the fourth-order lateral-motion case there would be: twelve variables to be manipulated, four soft constraint equations to be approximately satisfied, n hard constraint equations where n is the sum of the specified pole-zero excesses, and six envelopes to be matched.

One simplification which circumvents the need for a cost functional and optimization strategy is the use of direct search. If any solution which satisfies the constraints and the envelopes is as useful as any other, systematic or random direct search is easier to implement and no more wasteful of computer time than optimization. Random direct search has been used to obtain second-order longitudinal airplane transfer functions. In this simpler situation the $a_{ij} : i, j = 1, 2$ are randomly selected. If the resulting eigenvalues fall within the region shown in Figure 5 [4], the $b_i : i = 1, 2$ are randomly selected and C^* and $\hat{C}^*$ calculated. The ranges of the random choices are very

influential. Sample results are:

1. 558 random sets of a_{ij} yielded 22 acceptable pairs of real eigenvalues and six acceptable pairs of complex eigenvalues.
2. For $\underline{A}$ having acceptable real eigenvalues 118 random pairs of b_i were required to find one pair which yielded time-responses which match the $\dot{C}^*$ and $\ddot{C}^*$ envelopes.
3. For $\underline{A}$ having acceptable complex eigenvalues 320 random pairs of b_i were required to find one pair which satisfied the envelopes.
4. If one envelope was matched (and the other violated) it was as likely to be the $\dot{C}^*$ envelope as the $\ddot{C}^*$ envelope.

The application of direct search to the fourth-order problem is currently being investigated. This is potentially useful for the development of models which satisfy the longitudinal criteria, $\dot{C}^*$ and $\ddot{C}^*$, but less so for the lateral case. In order to develop lateral motion models and obviate the use of iterative numerical techniques the problem must be reposed.

If one begins with specific time responses which fall within the envelopes and seeks a model whose output responses closely resemble the specified handling-quality time histories the problem no longer requires iteration. If the eigenvalues are specified or obtained from the input responses and then frozen and if all elements of $\underline{A}$ and $\underline{b}$ are unspecified rather than just those elements which are normally nonzero for most airplanes, the problem is well-posed. This procedure has three disadvantages:

1. The partial degrees of freedom represented by loose bounds on

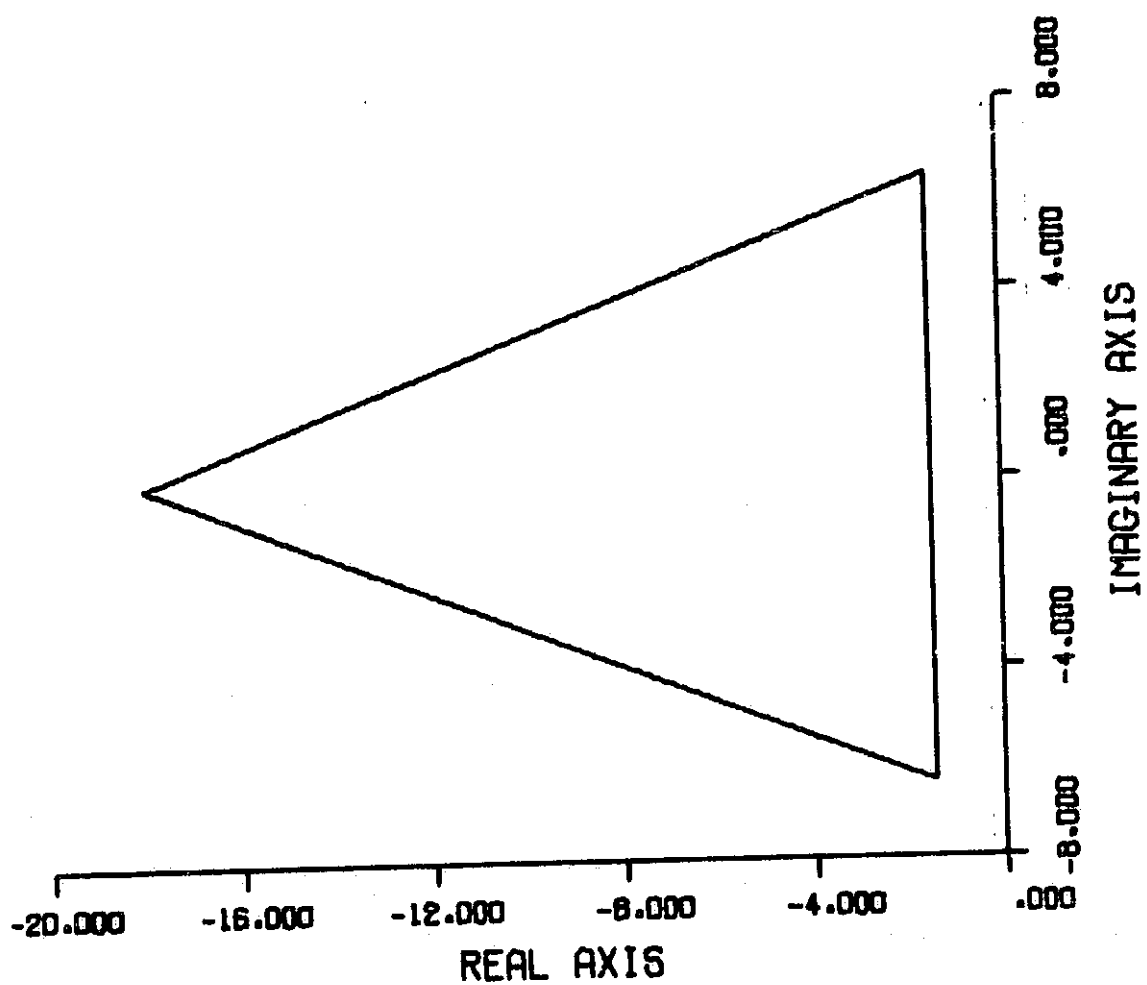


FIG.5 ACCEPTABLE EIGENVALUES

on the eigenvalues are lost.

2. The envelopes are not directly used and may be violated by the results.
3. The resulting $\underline{A}$ is not constrained to have the zero elements and kinematic elements usually found in such models of airplanes.

Experience with lateral-motion examples has shown that $\underline{A}$ may well have obviously unrealistic elements and still produce realistic transfer functions. This is sufficient for present purposes.

The reposed problem is the following: given $\hat{y}(t)$, $\underline{h}(a_{ij})$ and $\underline{G}(a_{ij})$, find $\underline{A}$ and $\underline{b}$ such that if

$$d\underline{x} = \underline{A}\underline{x} + \underline{b}\delta_a, \quad (6)$$

and if

$$\underline{y} = \underline{G}\underline{x} + \underline{h}\delta_a, \quad (7)$$

then

$$\underline{y} \approx \hat{\underline{y}} \text{ in some best sense,}$$

with

$$\underline{\Lambda} = \hat{\underline{\Lambda}} \quad (8)$$

Note that the elements of $\underline{G}$ and $\underline{h}$ are known functions of the unknown elements of $\underline{A}$ and that $\underline{G}$ is normally not square, there being fewer elements in $\underline{y}$ than in $\underline{x}$.

The first step is to obtain an analytical representation of the input time histories which one specifies in the form of a table of discrete values uniformly spaced in time. If the model is to be a fourth-order constant-

coefficient linear ordinary differential equation, its solution must have the form:

$$y_1(t) = c_{11}e^{\lambda_1 t} + c_{12}e^{\lambda_2 t} + c_{13}e^{\lambda_3 t} + c_{14}e^{\lambda_4 t} + c_1 \quad (9)$$

These coefficients and eigenvalues, c_{ij} and λ_j , are obtained by least-squared-error fitting the above form to the specified discrete values of $\hat{y}_1$. This is a two-step process. First the λ_j are calculated. The calculation is based upon first and second differences between adjacent $\hat{y}_1$ values and is strongly influenced by their precision. If the values are represented by three significant figures the resulting eigenvalues are drastically incorrect even when the data are contrived and there should be an exact solution. As the number of significant figures is increased the calculated eigenvalues more closely resemble the correct values. However, if data obtained from sketched time responses or imprecise tables are to be accommodated, it is impractical to calculate eigenvalues from such data. In such cases, the eigenvalues which the model is to have must be specified. In either case, the c_{ij} are straightforwardly calculated and the resulting fits usually quite good. Thus one can represent the input time histories, $\hat{y}(t)$, as

$$\hat{y} \approx \underline{y} = \underline{C}e^{\underline{\lambda}t} + \underline{c} \quad (10)$$

where the elements of $\underline{c}$ are the c_1 .

An example set of discrete input data points and their fitted representations is shown in Figures 6 through 8. Reasonable responses were sketched onto graphs of the lateral handling-quality envelopes and then converted to tabular form by estimating the values at half-second intervals. The eigenvalues used in the curve-fitting process were specified. The values used were:

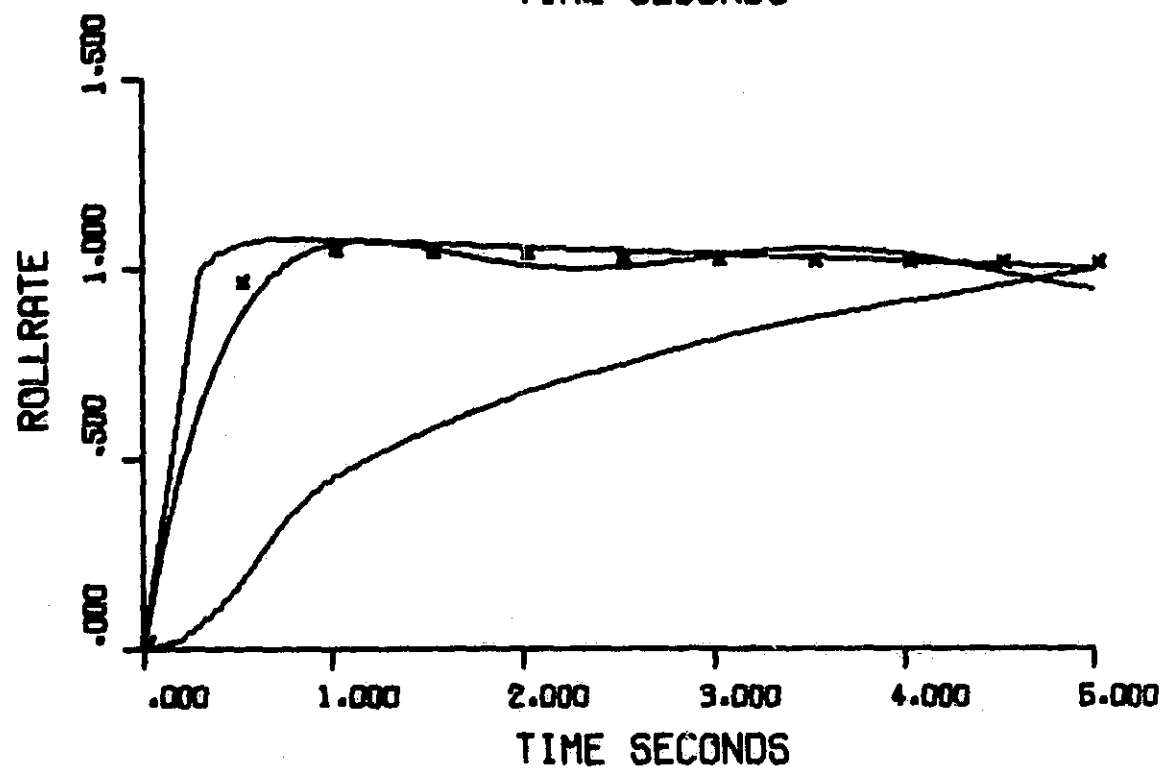
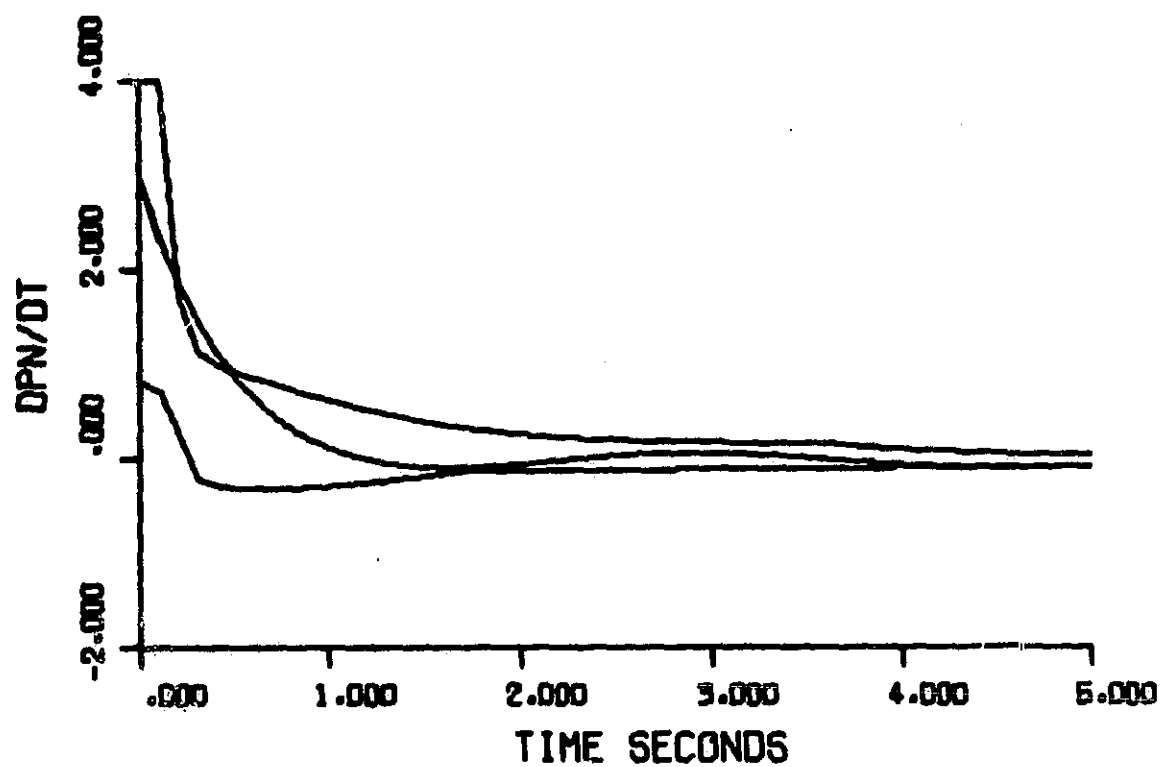


FIG.6 FITTED ROLLRATE RESPONSE

ORIGINAL PAGE IS
OF POOR QUALITY

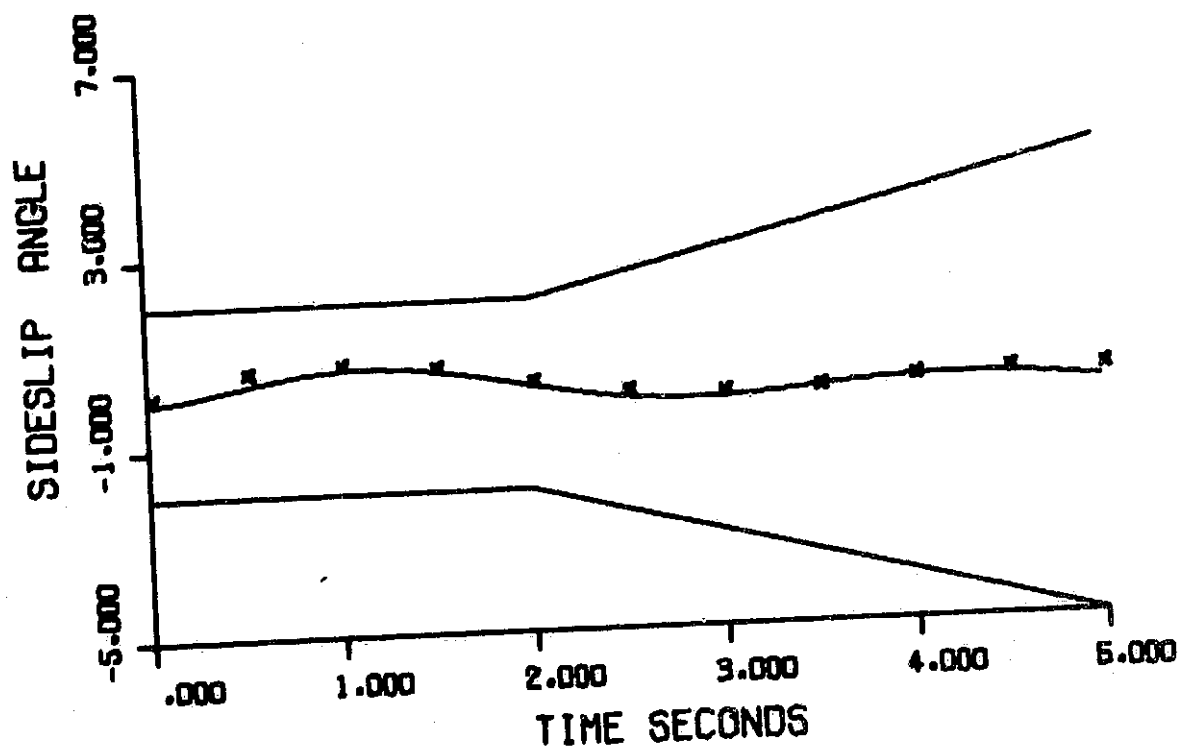
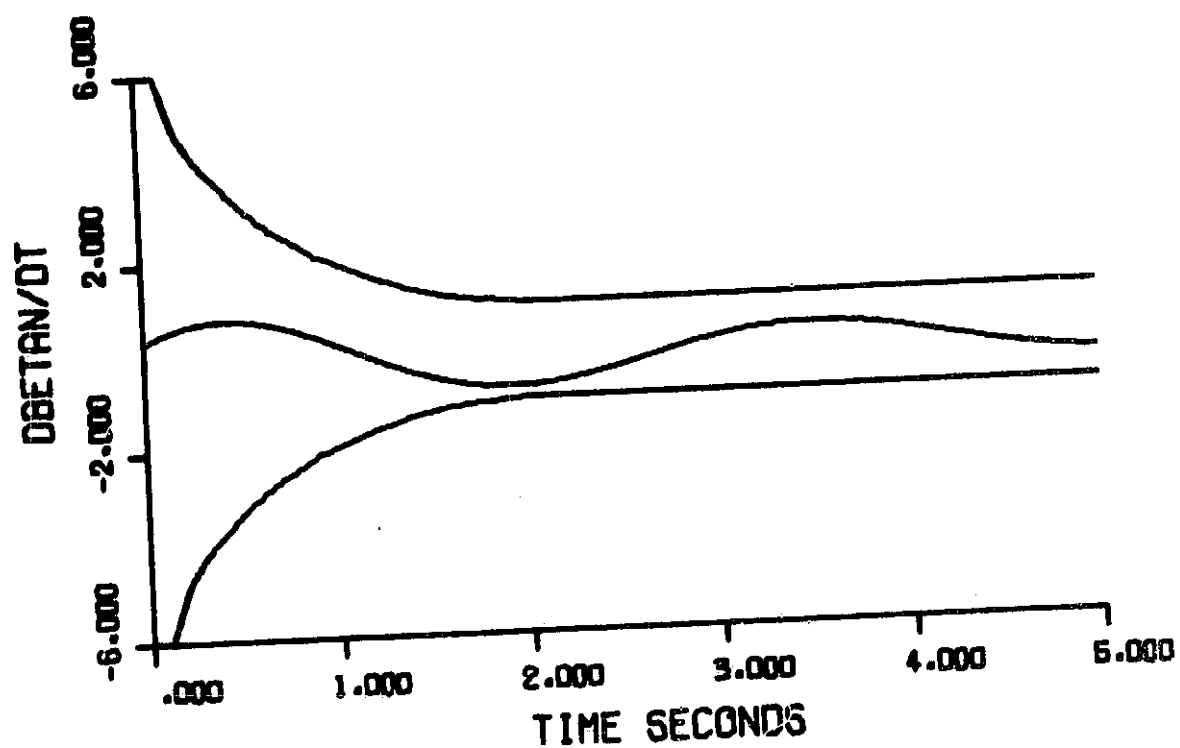


FIG.7 FITTED SIDESLIP RESPONSE

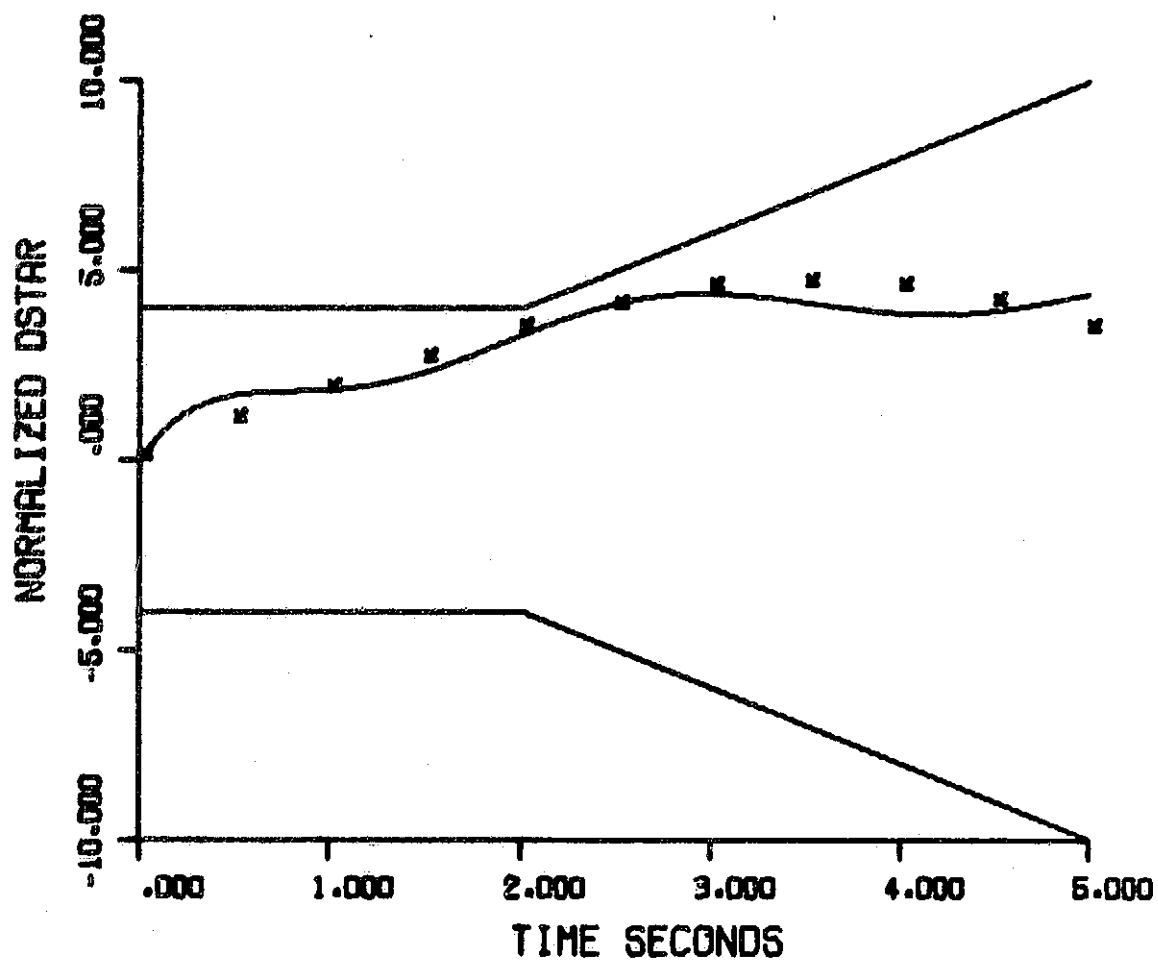


FIG.8A FITTED DSTAR RESPONSE

ORIGINAL PAGE IS
OF POOR QUALITY

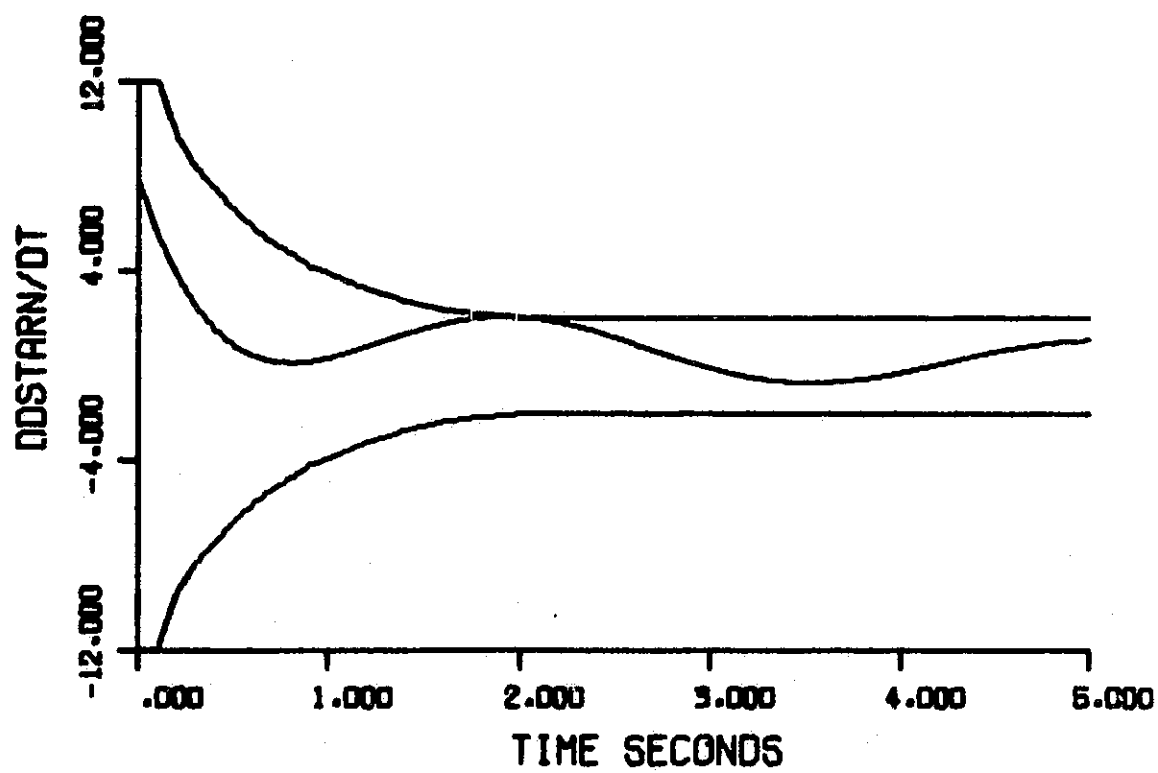


FIG. 8 B

$$\lambda_1 = -2.4$$

$$\lambda_2 = -.003$$

$$\lambda_3 = -.25 - j2.0$$

$$\lambda_4 = -.25 + j2.0$$

Once the specified discrete values of the $\hat{y}_i$ are approximately represented

$$\underline{y} = \underline{c}e^{\lambda t} + \underline{c}$$

This equation can be Laplace transformed into

$$\underline{Y}(s) = \frac{\underline{N}(s)}{\underline{D}(s)} \quad (11)$$

Assuming a unit-step input, the plant and output equations can also be Laplace transformed to

$$s\underline{X}(s) = \underline{A}\underline{X}(s) + \underline{b}/s$$

$$\underline{Y}(s) = \underline{G}\underline{X}(s) + \underline{h}/s$$

which are manipulated to obtain

$$\underline{X}(s) = (s\underline{I} - \underline{A})^{-1}\underline{b}/s = \underline{\phi}(s)\underline{b}/s$$

$$\underline{Y}(s) = \underline{G}\underline{\phi}(s)\underline{b}/s + \underline{h}/s \quad (12)$$

If equation 12 is rearranged to match the form of equation 11, the right-hand sides of equations 11 and 12 can be equated, coefficient by coefficient, to produce $n(m+1)$ simultaneous algebraic equations where n is the order of the plant equation, equation 6, and m is the number of elements in the output vector, equation 7. For low-order models this is a satisfactory technique.

The algebraic equations are of degree $\leq n$. In the fourth-order lateral-motion example there are sixteen nonlinear equations containing, at worst, quadruple-cross-product terms. A limited amount of experience with these equations has shown them to be numerically sensitive. An effort is being made to develop a computer program which will solve the fourth-order example equations with various constraints on the elements and eigenvalues of $\underline{A}$. While this may prove to be the most satisfactory approach to the problem, the numerical uncertainties are sufficient to motivate the development of an alternative technique which is computationally simpler.

A less numerically-sensitive approach can be developed by combining equations 7 and 10.

$$\underline{y} = \underline{G}\underline{x} + \underline{h}\delta_a$$

$$\underline{x} = \underline{G}^{-1}\underline{y} - \underline{G}^{-1}\underline{h}\delta_a$$

$$\underline{x} = \underline{G}^{-1}\underline{C}\underline{e}^{\lambda t} + \underline{G}^{-1}\underline{c} - \underline{G}^{-1}\underline{h}\delta_a$$

$$d\underline{x} = \underline{G}^{-1}\underline{C}\underline{A}\underline{e}^{\lambda t} - \underline{G}^{-1}\underline{h}d\delta_a$$

If these expressions are substituted into equation 6 one obtains

$$\underline{G}^{-1}\underline{C}\underline{A}\underline{e}^{\lambda t} - \underline{G}^{-1}\underline{h}d\delta_a = \underline{A}\underline{G}^{-1}\underline{C}\underline{e}^{\lambda t} + \underline{A}\underline{G}^{-1}\underline{c} - \underline{A}\underline{G}^{-1}\underline{h}\delta_a + \underline{b}\delta_a$$

or

$$(\underline{G}^{-1}\underline{C}\underline{A} - \underline{A}\underline{G}^{-1}\underline{C})\underline{e}^{\lambda t} = \underline{A}\underline{G}^{-1}\underline{c} + \underline{G}^{-1}\underline{h}d\delta_a + (\underline{b} - \underline{A}\underline{G}^{-1}\underline{h})\delta_a$$

For this expression to be true

$$\underline{G}^{-1}\underline{C}\underline{A} = \underline{A}\underline{G}^{-1}\underline{C} \quad (13a)$$

and

$$(\underline{A}\underline{G}^{-1}\underline{h} - \underline{b})\delta_a = \underline{A}\underline{G}^{-1}\underline{c} \quad (13b)$$

If $\underline{I} = \underline{G}^{-1}\underline{c}$, then

$$\underline{A} = \underline{I}\underline{A}\underline{I}^{-1} \quad (14)$$

and

$$\underline{b} = \underline{A}\underline{G}^{-1}(\underline{h} - \underline{c}) \quad (15)$$

It is assumed that $\delta_a(t) \equiv 1$ and $\underline{G}^{-1}\underline{h}\delta_a$ is neglected at this point but must be introduced as an initial condition vector when computing model time responses. A model obtained from equations 14 and 15 will have the specified eigenvalues and match the analytical representations of the input data exactly.

Unfortunately the above results assume the existence of $\underline{G}^{-1}$ whereas, in general, $\underline{G}$ is not square. One can substitute the right pseudoinverse and obtain similar results except that $\underline{I}$ will be singular and the model eigenvalues are no longer equal to the specified eigenvalues since $\underline{A}$ and $\underline{\Lambda}$ are no longer similar. If $\underline{G}$ is square and nonsingular, equations 14 and 15 hold. There are two cases of practical interest in which $\underline{G}$ is nonsingular. First, one can specify $n = m$, that is, let the number of time histories establish the order of the model. Secondly, it may be possible to adjoin created pseudo-time-histories to $\underline{y}$ until $m = n$. One must have the means to create these pseudo-time-histories. They must be linear combinations of the model state variables and the resulting distribution matrix must be nonsingular.

In the lateral-direction, small-motion example, the state variables are roll rate, yaw rate, sideslip angle and roll angle. The handling-quality indicators are normalized roll rate, normalized sideslip angle and normalized $\dot{D}^*$.

The normalized roll-rate input data can be integrated to produce normalized roll-angle pseudodata. For the example, this was accomplished by exactly fitting a tenth-order polynomial to the eleven discrete roll-rate data points shown in Figure 6. This polynomial was integrated and evaluated at the same values of time to generate the discrete pseudodata points shown in Figure 9. The continuous curve in Figure 9 results from fitting an equation of the form of equation 9 to the discrete pseudodata points just as is done for the normal data. This allows the expansion of $\underline{y}$ to $\tilde{\underline{y}}$ and $\underline{G}$ to $\tilde{\underline{G}}$ and assures the non-singularity of $\tilde{\underline{G}}$. This, in turn, allows calculation of $\underline{A}$ and $\underline{b}$ using equations 14 and 15. The same straightforward opportunity to augment $\underline{y}$ until $\tilde{\underline{G}}$ is square and nonsingular does not exist in the longitudinal small-motion case, although a designer could impose specifications on the motion, in addition to the Boeing C^* criterion, and thereby create a nonsingular $\tilde{\underline{G}}$.

COMPUTATIONAL PROCEDURES

D^* is a weighted combination of aircraft sideslip which is considered the principal low-speed handling-quality parameter and lateral acceleration at the pilot station which is the principal motion cue parameter at high speeds [4]. In terms of a stability-axis system representation of small perturbations about straight and level flight for a normally configured aircraft with the pilot station on the longitudinal principal axis the expression for D^* is

$$D^*(t) = V(\dot{\beta} + r) + \ell \dot{r} + c_3 q_{co} \beta \quad (16)$$

where c_3 is a dimensional constant defined in Table 1. In terms of the

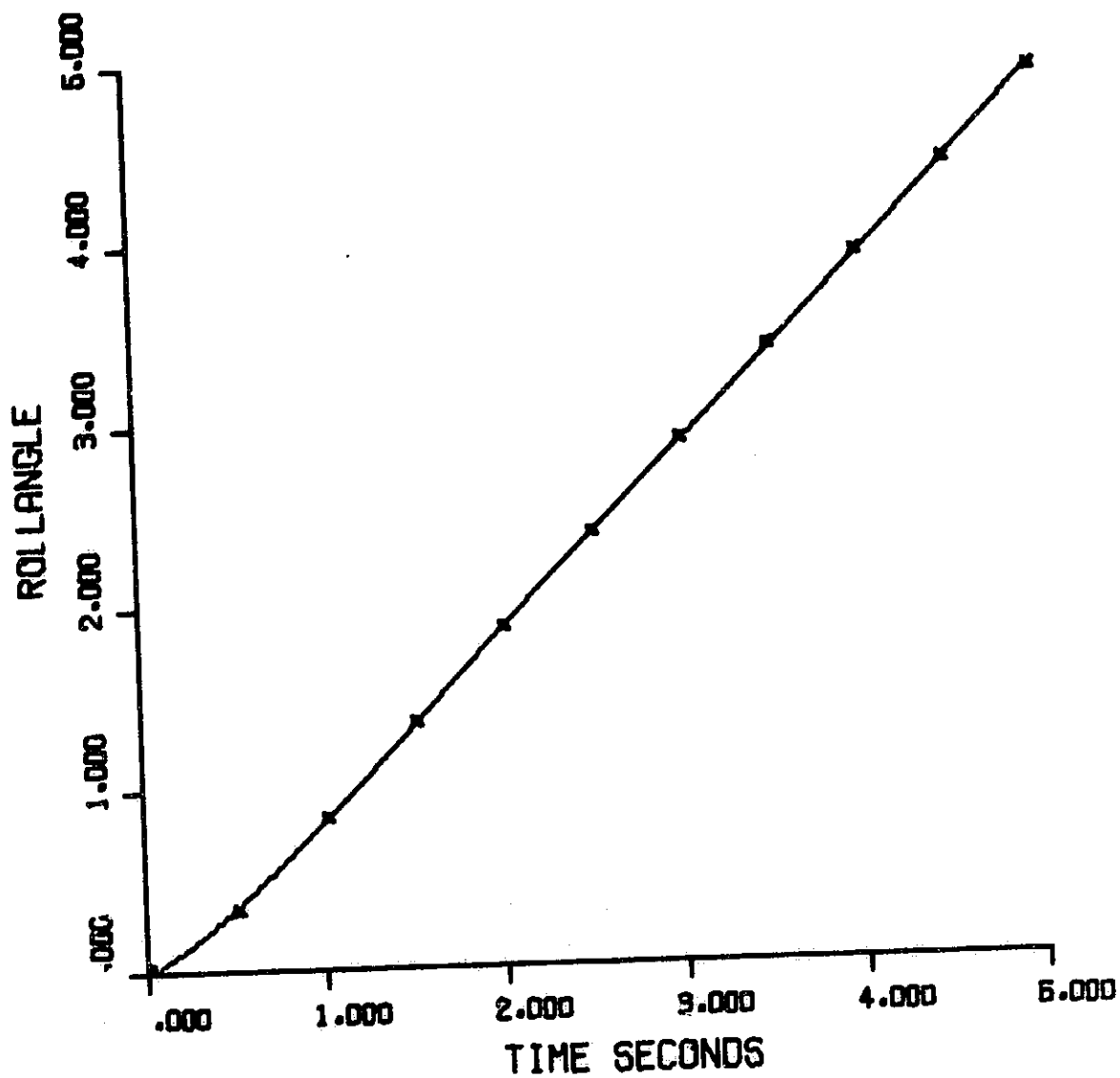


FIG.9 ROLL ANGLE PSEUDODATA

ORIGINAL PAGE IS
OF POOR QUALITY

elements $\underline{A}$ and $\underline{b}$ one obtains

$$D_p = Va_{31} + la_{21}$$

$$D_r = Va_{32} + la_{22} + V$$

$$D_\beta = Va_{33} + la_{23} + c_3 q_{co}$$

$$D_\phi = Va_{34} + la_{24}$$

$$D_{\delta_a} = Vb_3 + lb_1$$

and

$$D^*(t) = D_p p(t) + D_r r(t) + D_\beta \beta(t) + D_\phi \phi(t) + D_{\delta_a} \delta_a \quad (18)$$

Unit Correction Factors for D^* Equation

D^*	β	C_3	
		Value	Units
g's	rad	-9.91×10^{-3}	$\left(\frac{g's - ft^2}{lb} \right)$
	deg	-1.73×10^{-4}	$\left(\frac{g's - ft^2}{lb - deg} \right)$
ft/sec ²	rad	-3.19×10^{-1}	$\left(\frac{ft^3}{lb-sec^2} \right)$
	deg	-5.57×10^{-3}	$\left(\frac{ft^3}{lb-sec^2-deg} \right)$

Units for crossover dynamic pressure, q_{co} ; lb/ft²

Table 1 [4]

Note that this assumes that the plant matrix is

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} & \dots \\ a_{21} & & \\ \vdots & & \end{bmatrix}$$

With no constraints on the elements a_{ij} . Thus $\underline{A}$ is specifically not constrained to be of the form

$$\underline{A} = \begin{bmatrix} L_p & L_r & L_\beta & 0 \\ N_p & N_r & N_\beta & 0 \\ \alpha_o & -1 & \gamma_\beta & \gamma_\phi \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (19)$$

similarly

$$\underline{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad \text{and not} \quad \begin{bmatrix} L_{\delta a} \\ N_{\delta a} \\ \gamma_{\delta a} \\ 0 \end{bmatrix}.$$

Thus the output distribution matrix $\underline{G}$ for the lateral motion case is of the form

$$\underline{G} = \begin{bmatrix} 1/p_{ss} & 0 & 0 & 0 \\ 0 & 0 & 1/\beta_{ss} & 0 \\ D_p/D_{ss} & D_r/D_{ss} & D_\beta/D_{ss} & D_\phi/D_{ss} \end{bmatrix} \quad (20)$$

where

$$y_1(t) = p(t)/p_{ss} = p_n(t)$$

$$y_2(t) = \beta(t)/\beta_{ss} = \beta_n(t)$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$y_3(t) = D^*(t)/D_{ss} = D_n^*(t) \quad (21)$$

and

$$\underline{h} = \begin{bmatrix} 0 \\ 0 \\ D_{\delta_a}/D_{ss} \end{bmatrix} \quad (22)$$

The augmented output distribution matrix is

$$\tilde{\underline{G}} = \begin{bmatrix} 1/p_{ss} & 0 & 0 & 0 \\ 0 & 0 & 1/\beta_{ss} & 0 \\ 0 & 0 & 0 & 1/p_{ss} \\ D_p/D_{ss} & D_r/D_{ss} & D_\beta/D_{ss} & D_\phi/D_{ss} \end{bmatrix} \quad (23)$$

which is nonsingular if $D_r \neq 0$. The augmented output vector, $\tilde{y}(t)$, is

$$\begin{aligned} \tilde{y}_1(t) &= p_n(t) \\ \tilde{y}_2(t) &= \beta_n(t) \\ \tilde{y}_3(t) &= \int p_n(t)dt = \phi_n(t) \\ \tilde{y}_4(t) &= D_n^*(t) \end{aligned} \quad (24)$$

Since n^2 elements of $\underline{A}$ and the n elements of $\underline{b}$ are to be established equation 14 is equivalent to n^2 scalar equations with n^2 unknowns and equation 15 is equivalent to n scalar equations with n unknowns. This is not significantly different from the Laplace transform method in which one obtains $n(m+1)$ algebraic equations without having had to create psuedodata. However, the degree

of the simultaneous equations to be solved in the Laplace transform method is n whereas the equations arising from the pseudodata method contain cross-product terms at worst. This dramatically increases the likelihood of obtaining solutions by iterative numerical means. For the fourth-order lateral motion example, there are sixteen equations to be solved for the a_{ij} . A Newton-Euler procedure [5] was employed in which an initial estimate of the solution vector $\underline{a}$ is iteratively improved. If equation 14 is rewritten

$$\underline{f}(\underline{a}) = \underline{0} \quad (25)$$

and $\underline{P}(\underline{a})$ is the Jacobian matrix associated with the equation 25 then

$$\underline{a}_{k+1} = \underline{a}_k - \underline{P}^{-1}(\underline{a}_k)\underline{f}(\underline{a}_k) \quad (26)$$

This simple procedure has proved to be so successful that $\underline{A} = \underline{I}$ can be used as the starting point and every element of $(\underline{a}_{k+1} - \underline{a}_k)$ is reduced to 10^{-5} in three iterations typically. The elements of $\underline{b}$ are obtained from equation 15 without iteration.

Before $\underline{A}$ and $\underline{b}$ can be calculated, equations of the form of equation 10 must be fitted to the input data. The resulting coefficient arrays $\underline{C}$ and $\underline{c}$ appear in equations 14 and 15. Note that $\underline{I} = \underline{\tilde{G}}^{-1}\underline{C}$ in the pseudodata method.

A set of eigenvalues can be obtained from each discretized input time history. If

$$e_z = y_i(z\Delta t) - \hat{y}_i(z\Delta t)$$

and

$$\mu_j = e^{\lambda_j \Delta t}$$

then

$$e_z = c_{11}\mu_1^z + c_{12}\mu_2^z + c_{13}\mu_3^z + c_{14}\mu_4^z + c_1 - \hat{y}_1(z\Delta t) \quad (27)$$

where

$$z = 1, 2, \dots (\text{number of discrete values } \hat{y}_1(z\Delta t)).$$

The c_{ij} and c_1 can be eliminated by linearly combining the e_z . One obtains a linear set of simultaneous algebraic equations

$$D'\underline{\mu} = \underline{d'} \quad (28)$$

where the elements of D' are first differences of the table of $\hat{y}_1(z\Delta t)$ values and the elements of $\underline{d'}$ are second differences. Unfortunately, if the $\hat{y}_1(z\Delta t)$ values are obtained from time-response sketches or are imprecise for any reason the μ_j calculated from equation 28 are useless. Frequently the μ_j obtained in such cases have negative values from which no λ_j can be calculated.

In lieu of eigenvalues obtained from the input data, it has been necessary, in all practical calculations, to use specified eigenvalues. In such cases the calculation of $\underline{C}$ is based on minimizing

$$E = \sum_{z=1}^{11} e_z^2$$

where the lateral handling-quality time histories typically span five seconds and are discretized by taking values every half second for a total of eleven values per time history. The initial value of $\tilde{y}(t)$ is forced to be zero by

$$c_1 = - \sum_{j=1}^4 c_{1j} \quad (29)$$

As a check on the entire computational process once an $\underline{A}$ and $\underline{b}$ have been

calculated, $\underline{G}$ and $\underline{h}$ can be calculated, $\underline{x}(t)$ obtained by integration and $\underline{y}(t)$ calculated. This $\underline{y}(t)$, obtained by integration, is plotted with the fitted $\underline{y}(t)$. They should be identical. The integration method is described in Takahashi, et. al., page 103 [6]. The solution matrix is represented by a series expansion containing p terms. The number of terms is specified by a recipe attributed to Paynter

$$\frac{1}{p!} (nq)^p e^{nq} \approx 0.001 \quad (30)$$

where n is the size of the $\underline{A}$ matrix and q is the largest element of $\underline{A}\Delta t$.

The example fitted and integrated handling-quality time histories are plotted in Figures 10, 11, and 12.

ORIGINAL PAGE IS
OF POOR QUALITY.

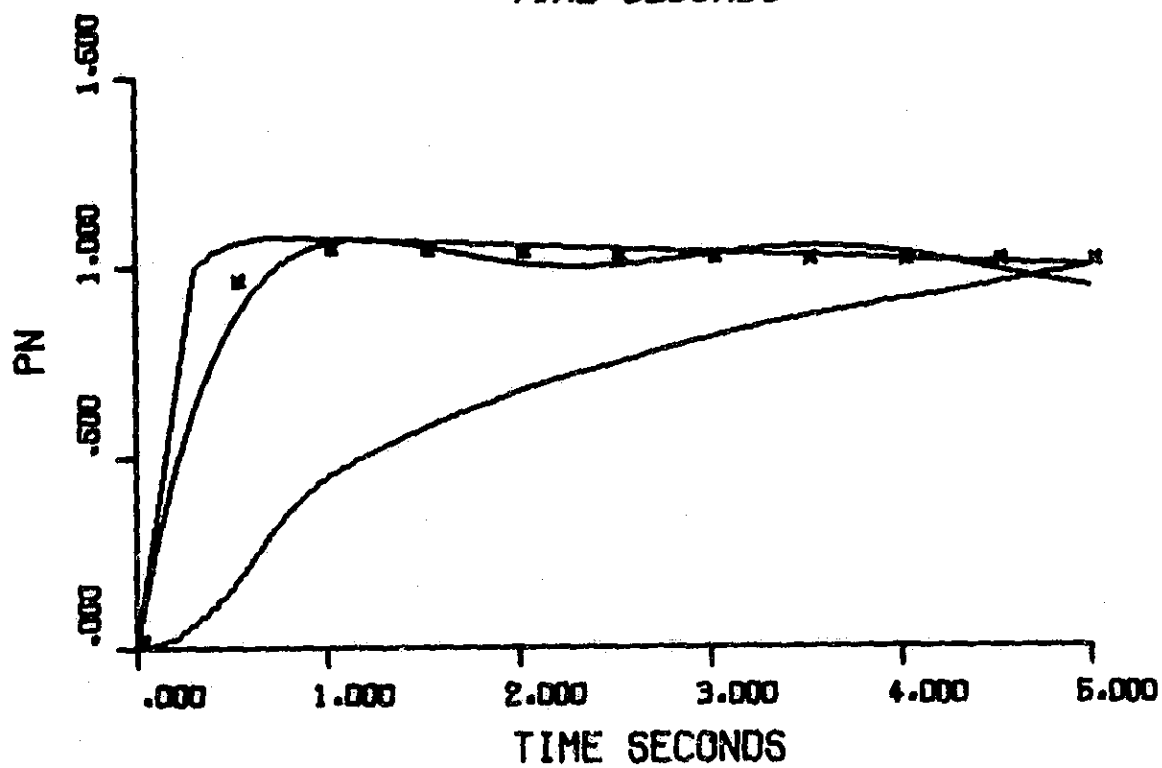
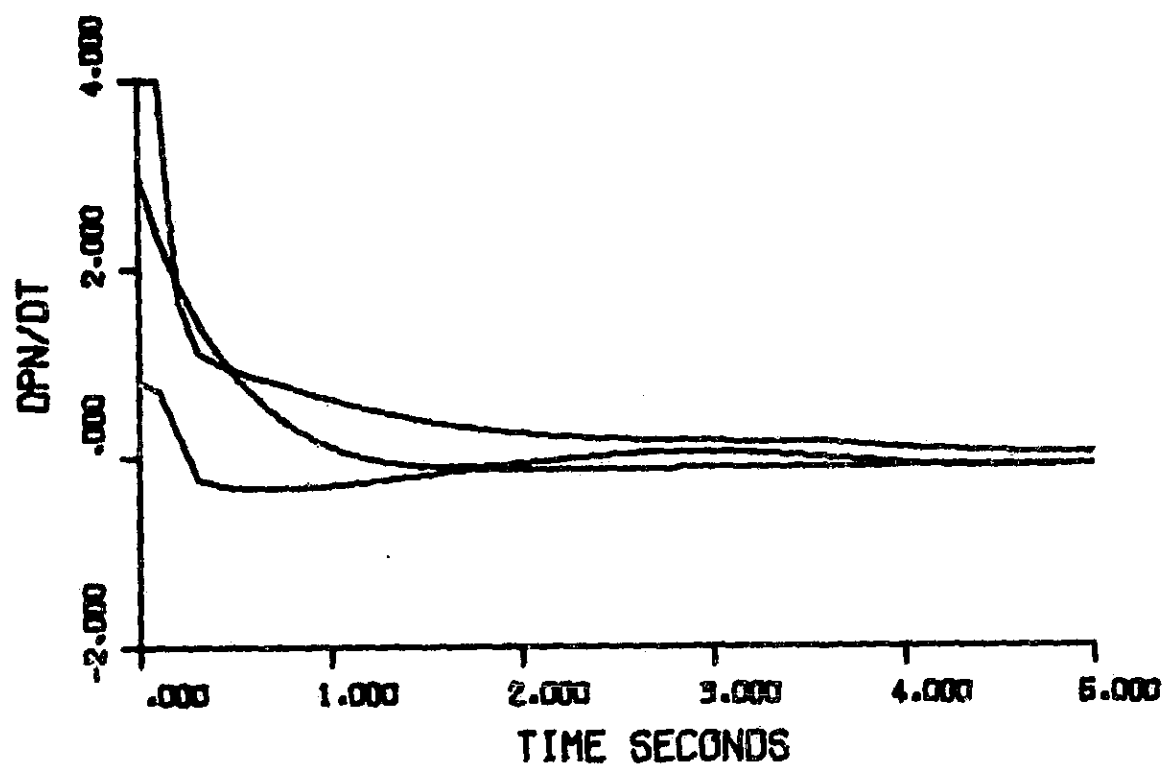


FIG.10 ROLLRATE

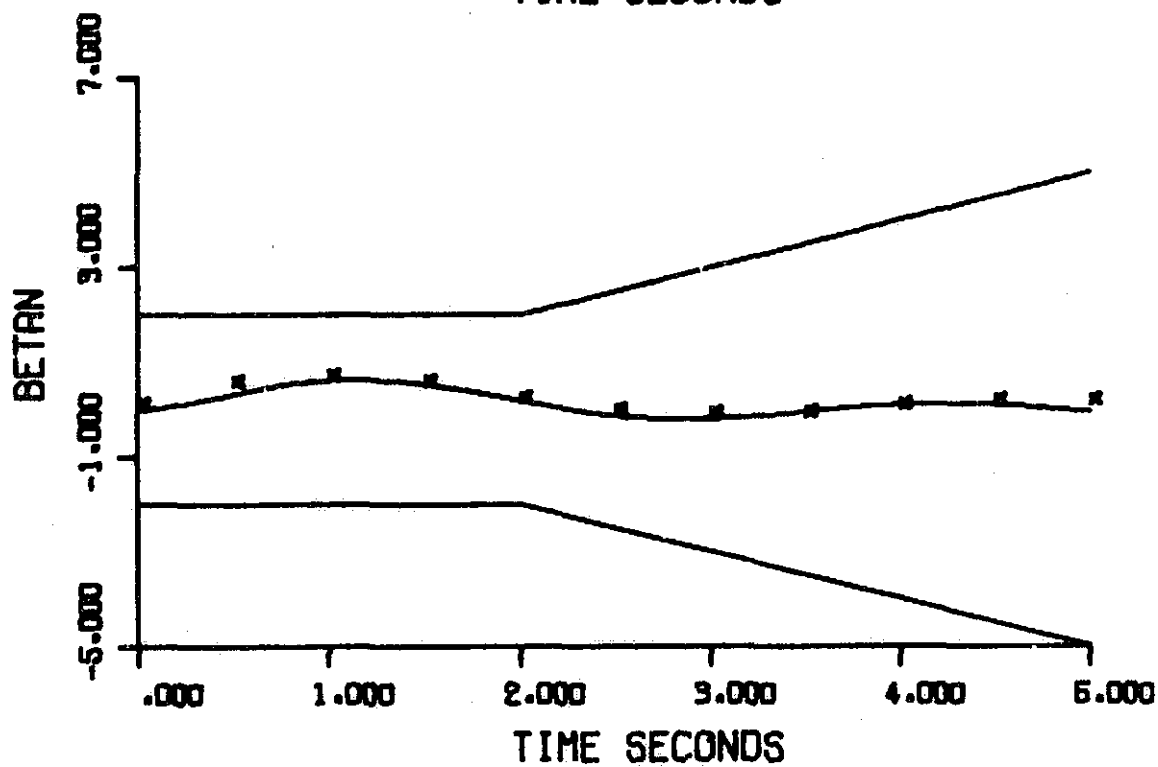
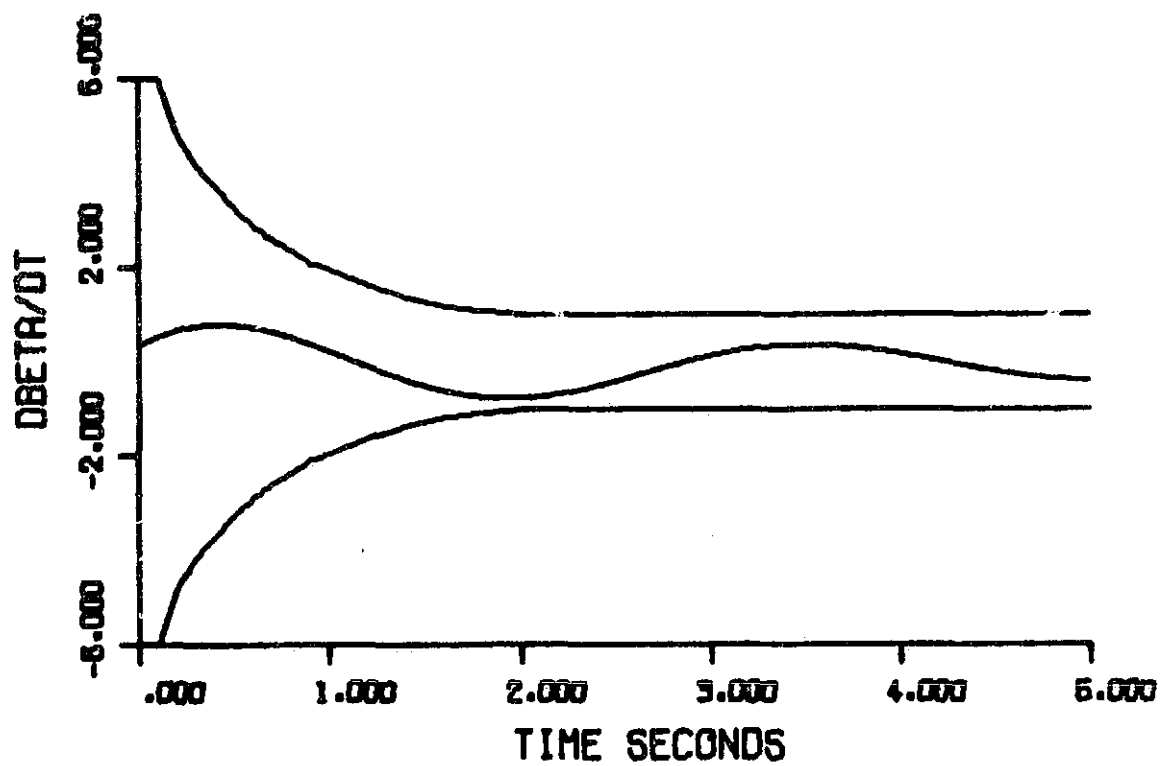


FIG.11 SIDESLIP

ORIGINAL PAGE IS
OF POOR QUALITY

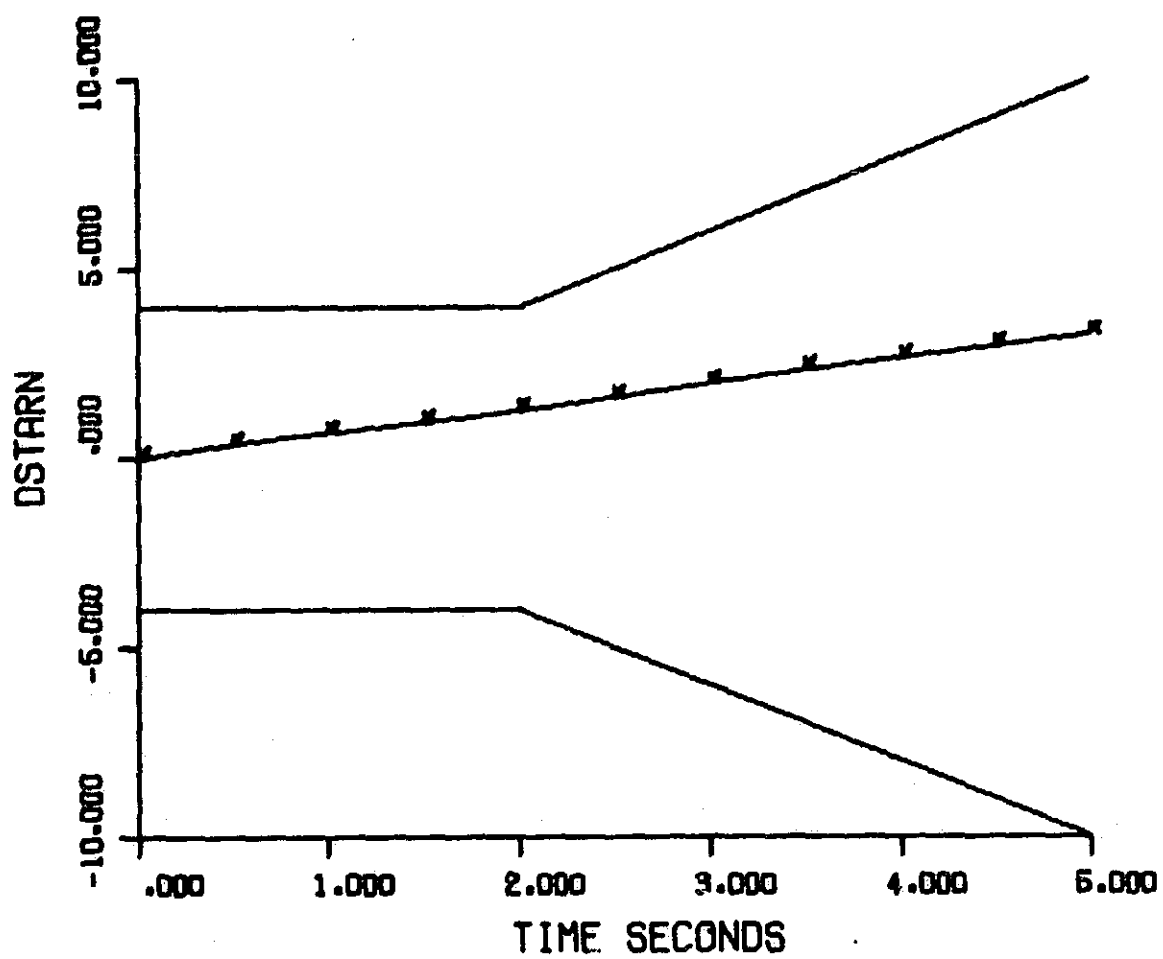


FIG.12 A LATERAL CRITERION

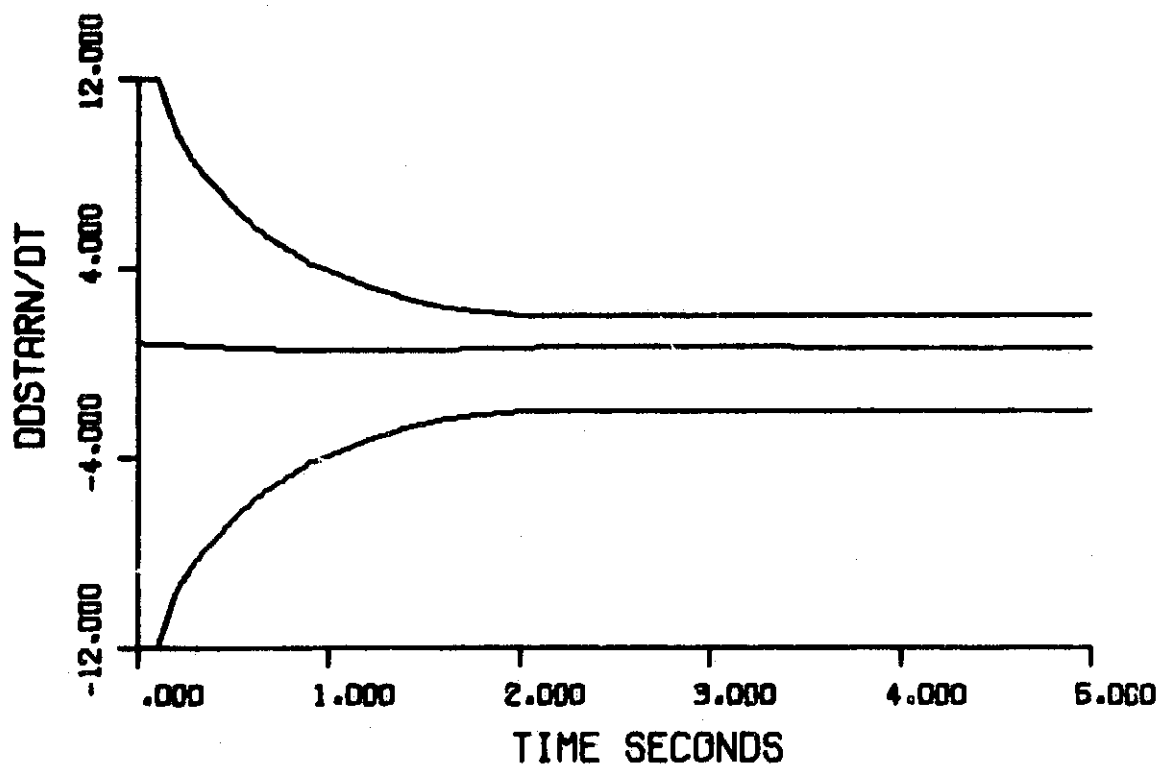


FIG. 12B

ZERO SUPPRESSION

The Laplace-transformed analytical representation of the input data, equation 11, has one fewer zero than poles in each element of the matrix transfer function $\underline{G}(s)$ where

$$\underline{Y}(s) = \underline{G}(s)/s$$

or

$$Y_i(s) = \frac{c_{i1}}{s-\lambda_1} + \frac{c_{i2}}{s-\lambda_2} + \frac{c_{i3}}{s-\lambda_3} + \frac{c_{i4}}{s-\lambda_4} + \frac{c_{0i}}{s}.$$

This expression can be rewritten, omitting the i subscript

$$\begin{aligned} & (c_1\lambda_1 + c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4)s^3 - (c_3\lambda_3\lambda_4 + c_4\lambda_3\lambda_4 + c_1\lambda_1\lambda_3 \\ & + c_3\lambda_1\lambda_3 + c_1\lambda_1\lambda_4 + c_4\lambda_1\lambda_4 + c_2\lambda_2\lambda_3 + c_2\lambda_2\lambda_4 + c_4\lambda_2\lambda_4 + c_1\lambda_2\lambda_1 \\ & + c_2\lambda_1\lambda_2)s^2 - (c_1\lambda_1\lambda_3\lambda_4 + c_3\lambda_1\lambda_3\lambda_4 + c_4\lambda_1\lambda_3\lambda_4 + c_4\lambda_1\lambda_3\lambda_4 + c_2\lambda_2\lambda_3\lambda_4 \\ & + c_3\lambda_2\lambda_3\lambda_4 + c_2\lambda_2\lambda_3\lambda_4 + c_1\lambda_1\lambda_2\lambda_3 + c_2\lambda_1\lambda_2\lambda_3 + c_3\lambda_1\lambda_2\lambda_3 \\ & + c_1\lambda_1\lambda_2\lambda_4 + c_2\lambda_1\lambda_2\lambda_4 + c_4\lambda_1\lambda_2\lambda_4)s + \lambda_1\lambda_2\lambda_3\lambda_4 \end{aligned} \quad (31)$$

To reduce the number of zeros in the transfer function, the following constraint equations must be incorporated into the least-square minimization, equation 29. To suppress one zero

$$c_1\lambda_1 + c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4 = 0 \quad (32)$$

To suppress two zeros

$$c_1\lambda_1 + c_2\lambda_2 + c_3\lambda_3 + c_4\lambda_4 = 0$$

and

$$\begin{aligned}
& c_3 \lambda_3 \lambda_4 + c_4 \lambda_3 \lambda_4 + c_1 \lambda_1 \lambda_3 + c_3 \lambda_1 \lambda_3 + c_1 \lambda_1 \lambda_4 + c_4 \lambda_1 \lambda_4 \\
& + c_2 \lambda_2 \lambda_3 + c_3 \lambda_2 \lambda_3 + c_2 \lambda_2 \lambda_4 + c_4 \lambda_2 \lambda_4 + c_1 \lambda_1 \lambda_2 + c_2 \lambda_1 \lambda_2 = 0 \quad (33)
\end{aligned}$$

If three zeros are suppressed, only one parameter remains to be established by the minimization and the resulting representation of the input time history is inadequate.

For low-order models, zero suppression has two detrimental effects on the models obtained. The first is the worsening of the agreement between the discretized input time histories and their analytical representations. For example, the unconstrained least-squared-error method yields a satisfactory fit of the roll-rate input, as shown in Figure 6. The continuous fitted curve has approximately the same relationship to the envelope as does the discretized input data. If the LSE method is constrained by equation 32 to have one less zero, one obtains the fit shown in Figure 13. If one further constrains the LSE minimization by equation 33, another zero will be eliminated and, for the example data, the fit is degraded to that shown in Figure 14. The same sequence is portrayed in Figures 7, 15, and 16 for the sideslip input. The Figure 7 results are unconstrained, one zero is suppressed in Figure 15 and two zeros are suppressed in Figure 16. Important features of the discrete inputs can be preserved by weighting the appropriate errors, e_z , in the LSE method. This may make the constrained fits more useful. Even the unconstrained cases can be altered. The risetime of the fitted roll rate can be reduced, for example. However, the overall fit cannot be improved by weighting.

The second detrimental effect of zero suppression is the increased tendency of the $\underline{A}$ matrix to contain unrealistic values. The $\underline{A}$ and $\underline{b}$ arrays resulting from unconstrained fits of the input time histories are

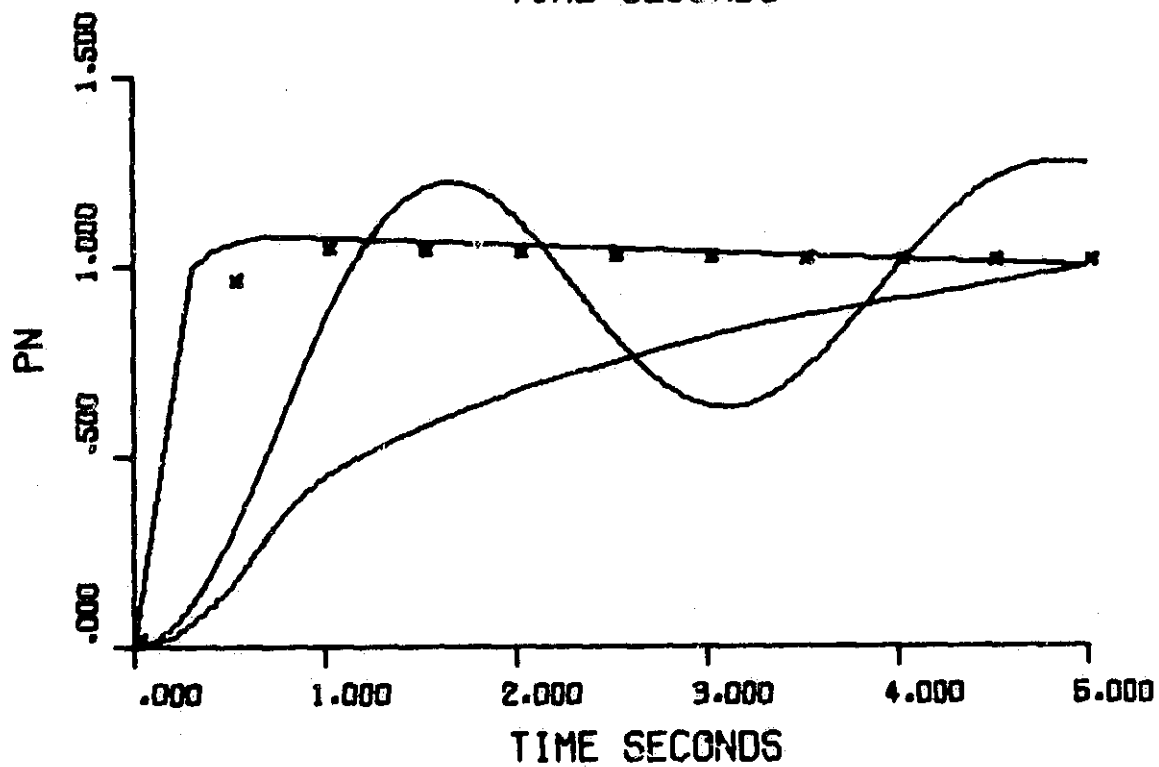
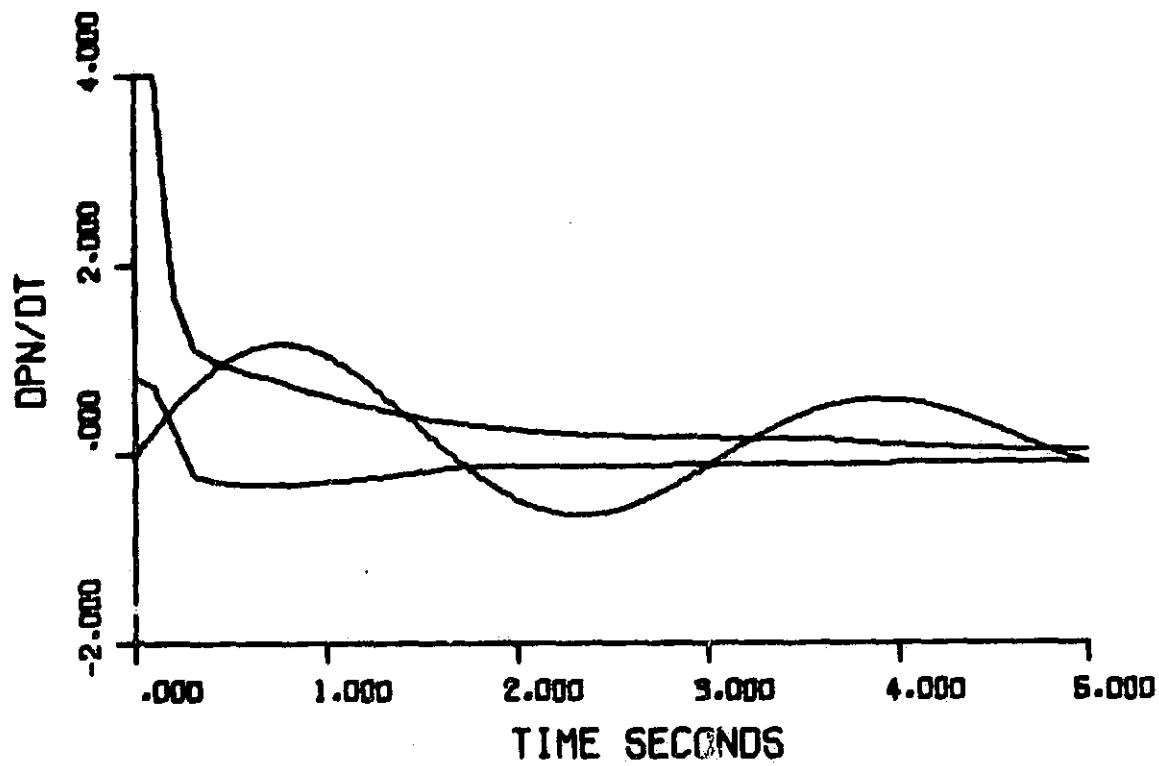


FIG.13 ROLLRATE

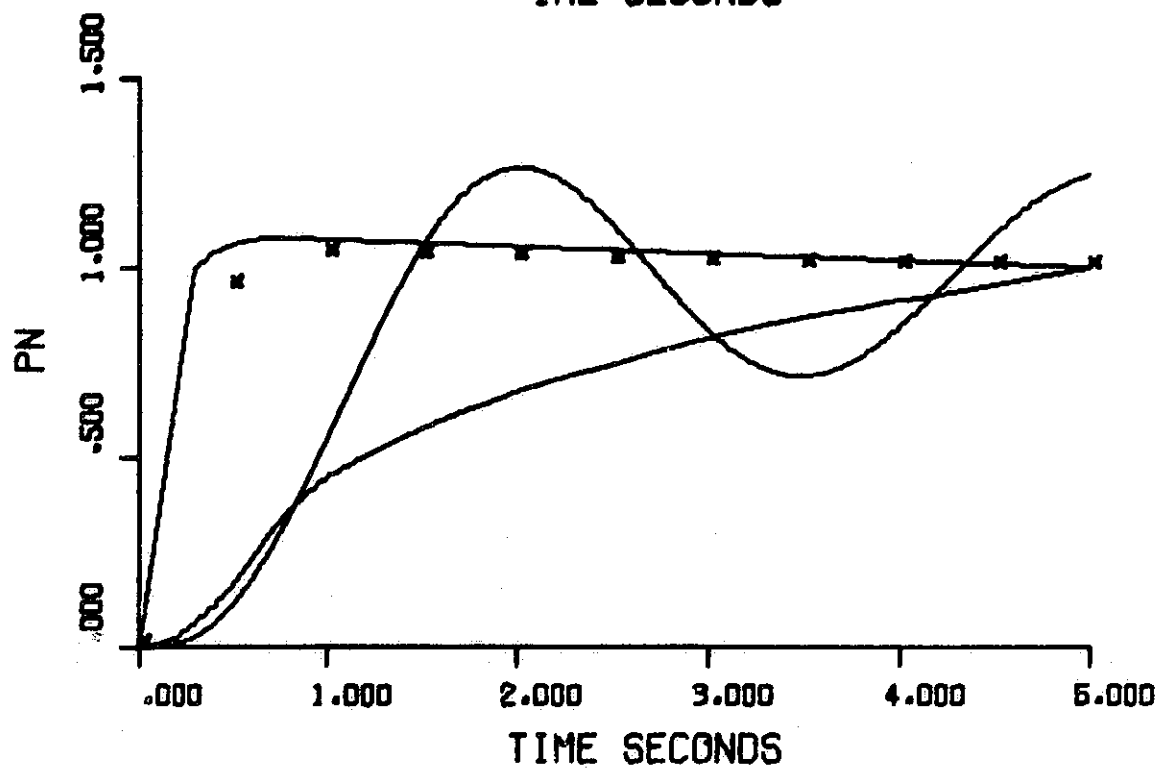
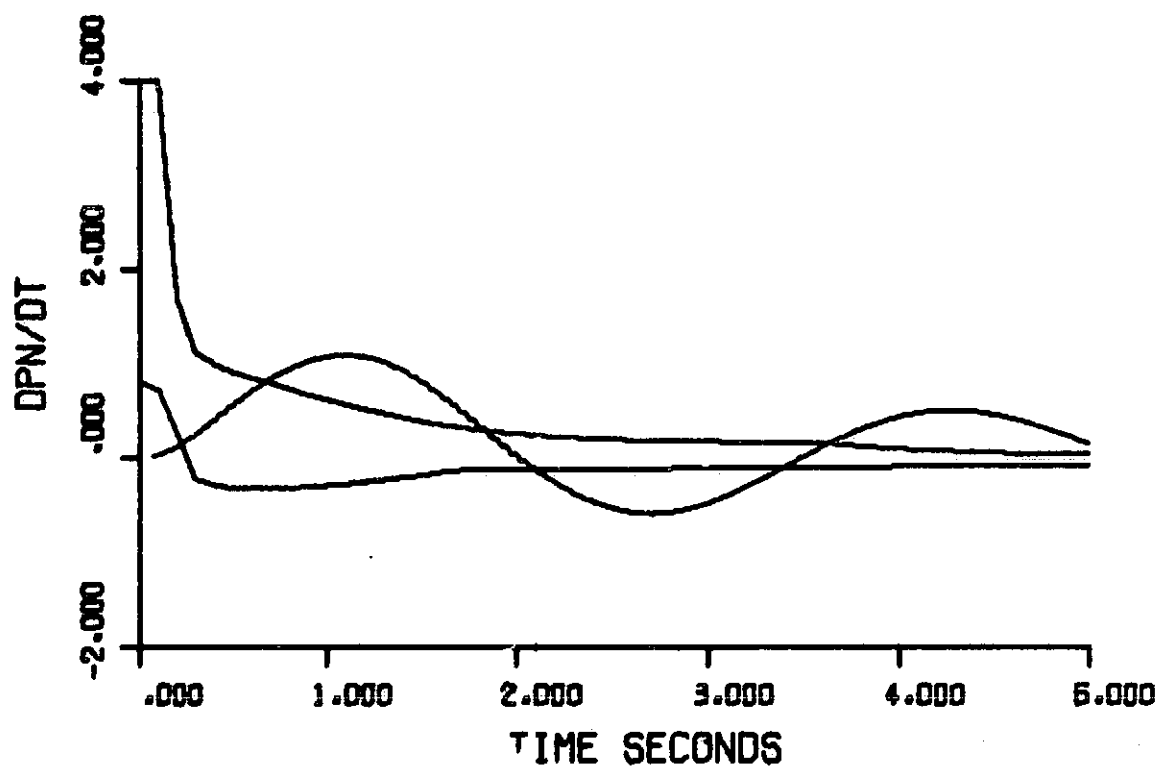


FIG.14 ROLLRATE

ORIGINAL PAGE IS
OF POOR QUALITY

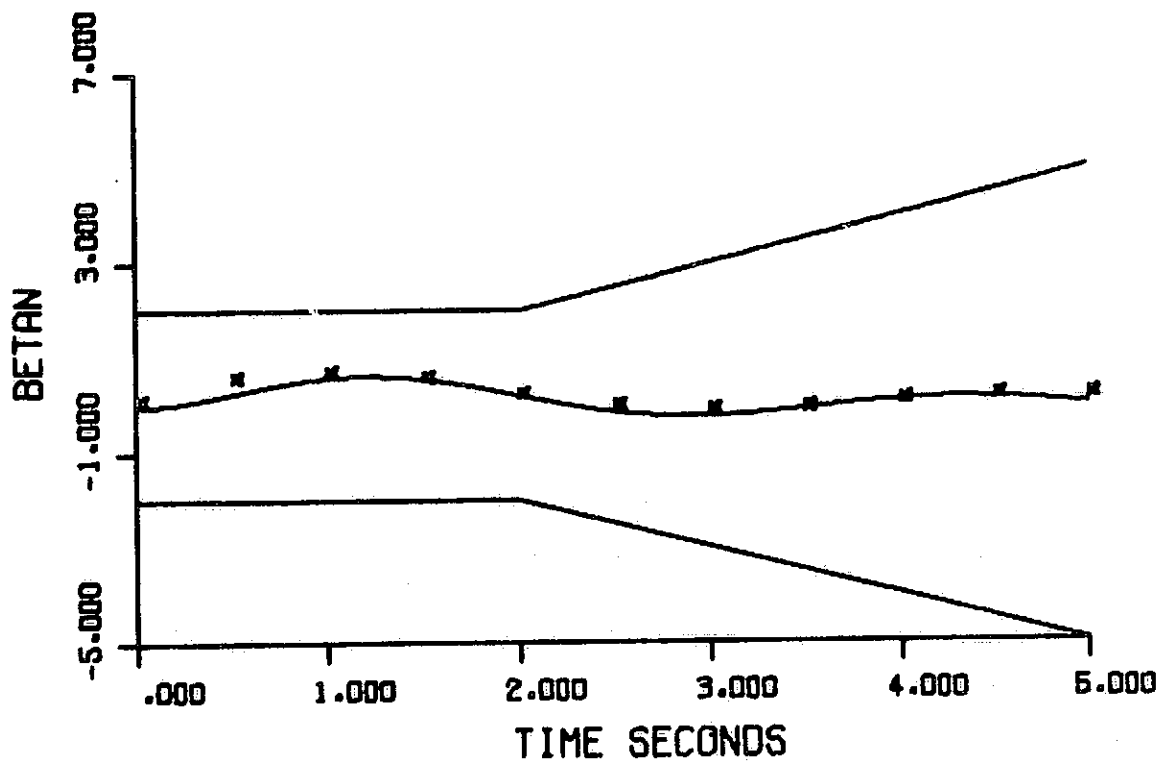
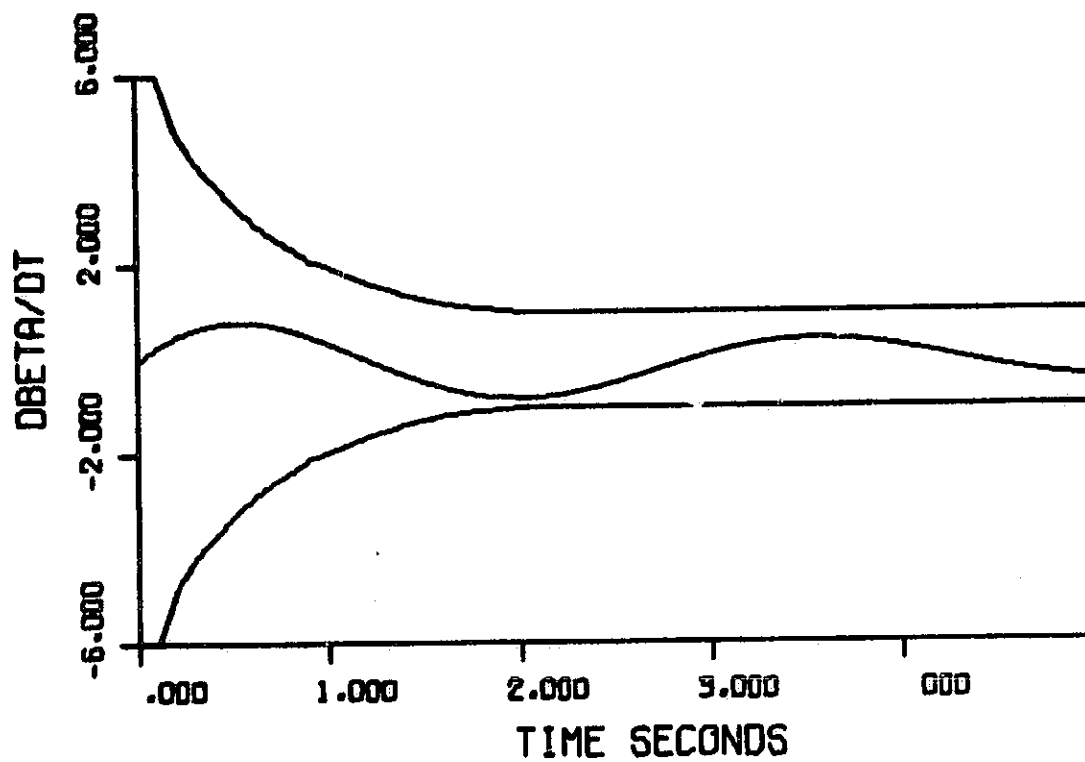


FIG.15 SIDESLIP

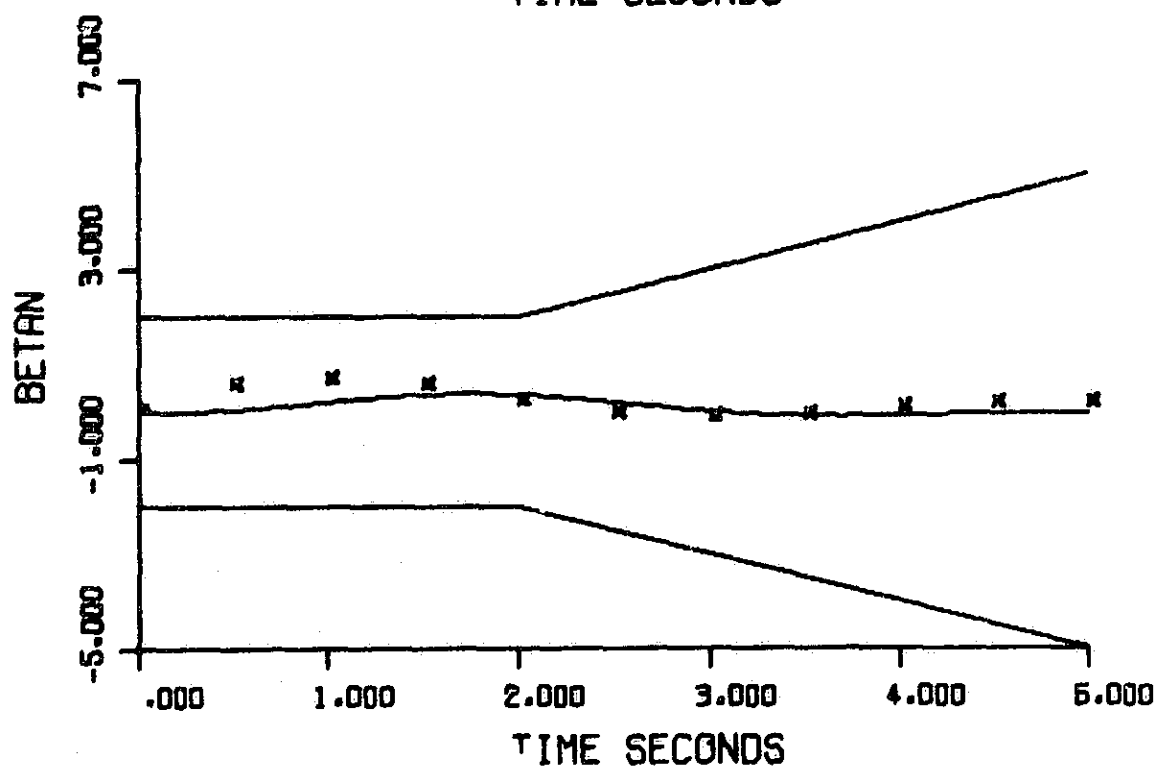
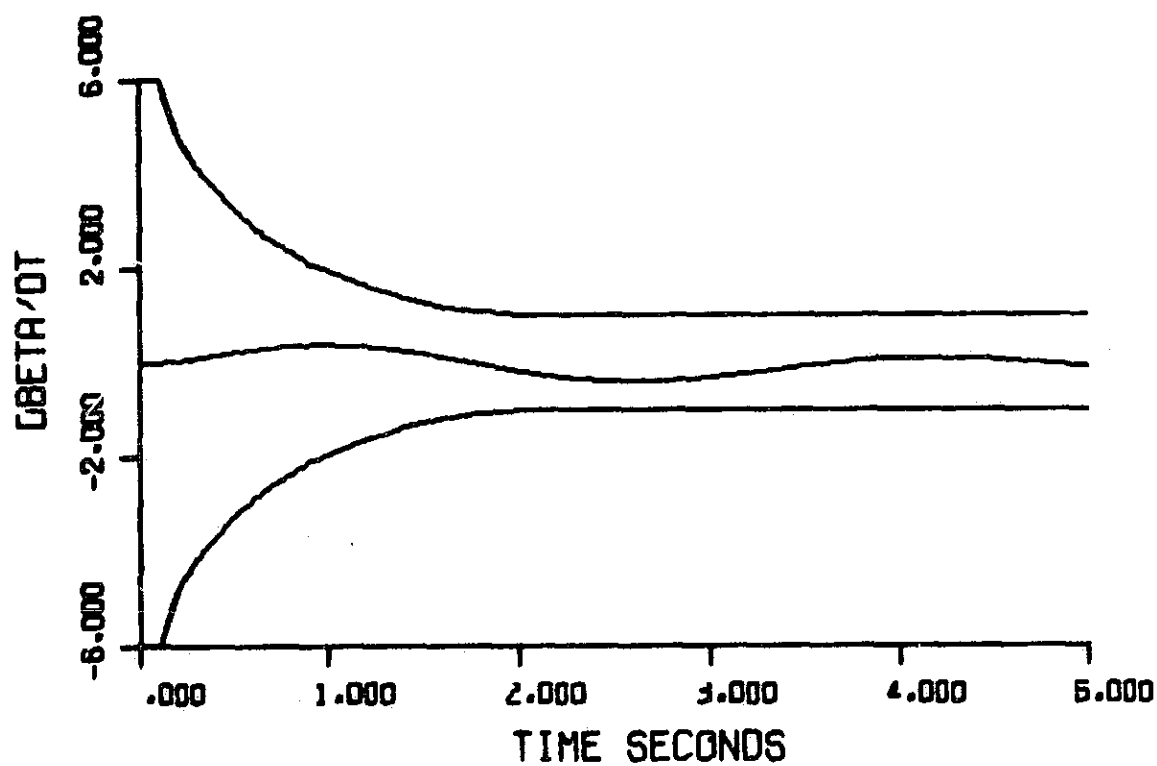


FIG.16 SIDESLIP

ORIGINAL PAGE IS
OF POOR QUALITY

$$\underline{A} = \begin{bmatrix} -1.36 & -4.17 & -20.22 & -.02 \\ -1.12 & 1.89 & 12.43 & -.04 \\ 0.22 & -.80 & -3.43 & 0.02 \\ 0.20 & 1.04 & 5.61 & 0.00 \end{bmatrix}$$

and

$$\underline{b} = \begin{bmatrix} 5.06 \\ 1.59 \\ -.12 \\ 1.09 \end{bmatrix}$$

The Newton-Euler procedure for obtaining the a_{ij} converged in three iterations to a maximum variation of any element from one iteration to the next of .00001 or less. This model was obtained from the unconstrained fits shown in Figures 6, 7, and 8.

If one zero in the roll-rate transfer function and one zero in the side-slip transfer function are suppressed, the $\underline{A}$ and $\underline{b}$ arrays are altered to

$$\underline{A} = \begin{bmatrix} -3.49 & 4.73 & 97.14 & 0.52 \\ 0.96 & -3.42 & -22.48 & -.07 \\ -.19 & 0.30 & 4.00 & 0.02 \\ -.32 & 2.44 & 15.42 & 0.01 \end{bmatrix}$$

and

$$\underline{b} = \begin{bmatrix} 0.00 \\ 1.06 \\ 0.00 \\ 0.87 \end{bmatrix}$$

The Newton-Euler procedure again required three iterations. This model was

obtained from the fits shown in Figures 13, 15, and 8.

If two zeros in the roll-rate transfer function and two zeros in the sideslip transfer function are suppressed, the $\underline{A}$ and $\underline{b}$ arrays are

$$\underline{A} = \begin{bmatrix} -9.34 & -1.66 & 403.88 & 2.75 \\ 3.51 & -2.88 & 138.62 & -.86 \\ -.21 & -.04 & 8.73 & 0.06 \\ -2.10 & 1.45 & 96.10 & 0.59 \end{bmatrix}$$

and

$$\underline{b} = \begin{bmatrix} 0.00 \\ 1.45 \\ 0.00 \\ 0.87 \end{bmatrix}$$

Three iterations were sufficient and the model was obtained from the data shown in Figures 14, 16, and 8.

The number of terms in the expansion of the state transition matrix specified by Paynter's recipe, equation 30, is a function of the largest absolute value contained in $\underline{A}$ and this number is frequently larger than the practical limit of about 30 when zeros are suppressed.

The additional information required to calculate $\underline{A}$ and $\underline{b}$ from the discrete time histories shown in Figures 7-16 is given in Table 2. The specified eigenvalues were

$$\lambda_1 = -2.4$$

$$\lambda_2 = -.003$$

$$\lambda_{3,4} = -.25 \pm j2.0$$

ORIGINAL PAGE IS
OF POOR QUALITY

AIRPLANE MODEL WITH SPECIFIED TIME HISTORIES

FLIGHT AND VEHICLE PARAMETERS

AIRSPEED	612.2 FT/SEC	
CG TO PILOT STATIC LONGITUDINAL DISTANCE		22.24 FT
DIMENSIONAL CONSTANT FOR DSTAR EQUATION		-.3190 CUBIC-FEET/LB-SECONDS-SQUARED
DYNAMIC PRESSURE	331.8 LB/FT-SQUARED	
ROLLRATE NORMALIZATION FACTOR	.500	
SIDESLIP NORMALIZATION FACTOR	10.000	
DSTAR NORMALIZATION FACTOR	.010	

Jetstar Parameters and Flight Conditions

Table 2

The eigenvalues and vehicle and flight parameters are approximately those of a Lockheed Jetstar, a four-engined utility transport [7], at 20,000 feet altitude and Mach = 0.6. It should be noted that the discrete input time histories were obtained from p_n , β_n , and D_n^* responses sketched by an FRC engineer [8] and do not refer to a particular airplane or flight condition. Finally, FRC program CONTROL was used to calculate transfer function coefficients from the above $\underline{A}$ and $\underline{b}$ arrays in combination with the appropriate distribution matrices, calculated from equation 22, to verify the suppression of the specified zeros.

VERIFICATION

Flight test data obtained from the FRC Lockheed Jetstar [9] provides a test case for the above model-generation procedure. For small, lateral variations about straight and level flight, the Jetstar can be represented by*

$$\underline{A} = \begin{bmatrix} -2.353 & 0.735 & -11.050 & 0.000 \\ -.057 & 0.358 & 3.836 & 0.000 \\ 0.026 & -.999 & -.205 & 0.053 \\ 1.000 & 0.054 & 0.000 & 0.000 \end{bmatrix}$$

$$\underline{b} = \begin{bmatrix} 5.650 \\ 0.031 \\ -.001 \\ 0.000 \end{bmatrix}$$

* Body axis nondimensional stability derivative parameters used in place of stability axis values for verification purposes only.

$$\tilde{\mathbf{G}} = \begin{bmatrix} 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \\ 14.75 & -7.044 & -146.0 & 32.14 \end{bmatrix}$$

$$\tilde{\mathbf{h}} = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

The eigenvalues were

$$\lambda_1 = -.24045$$

$$\lambda_2 = -.00310$$

$$\lambda_{3,4} = -.25428 \pm j2.06475$$

CONTROL was used to calculate the handling-quality time histories for the period 0.0 to 5.0 seconds. The discrete input data extracted from the CONTROL output is given in Table 3. The flight and vehicle parameters are given in Table 2. The normalization factors were arbitrarily selected.

The responses of the resulting model are shown in Figures 17, 18, and 20 and compared with the discrete input values of Table 3 after normalization. The $\mathbf{A}$, $\mathbf{b}$, $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{h}}$ arrays which define the model are

$$\mathbf{A} = \begin{bmatrix} -2.359 & 0.771 & -10.939 & -.003 \\ -.053 & -.359 & 3.849 & 0.000 \\ 0.025 & -1.005 & -.201 & 0.053 \\ 0.906 & -.021 & .278 & 0.003 \end{bmatrix}$$

Table 3

Time	Rollrate	Sideslip	D*
0.0	0.00	0.00	0.0
0.5	1.64	.016	37.6
1.0	2.04	.058	67.6
1.5	2.04	.093	95.1
2.0	2.01	.098	126.
2.5	2.06	.080	161.
3.0	2.14	.065	198.
3.5	2.18	.069	234.
4.0	2.15	.089	265.
4.5	2.09	.109	295.
5.0	2.05	.115	327.

These values are multiplied by the normalization factors listed in Table 2 before they are plotted.

Jetstar Discrete Time-History Data

**ORIGINAL PAGE IS
OF POOR QUALITY**

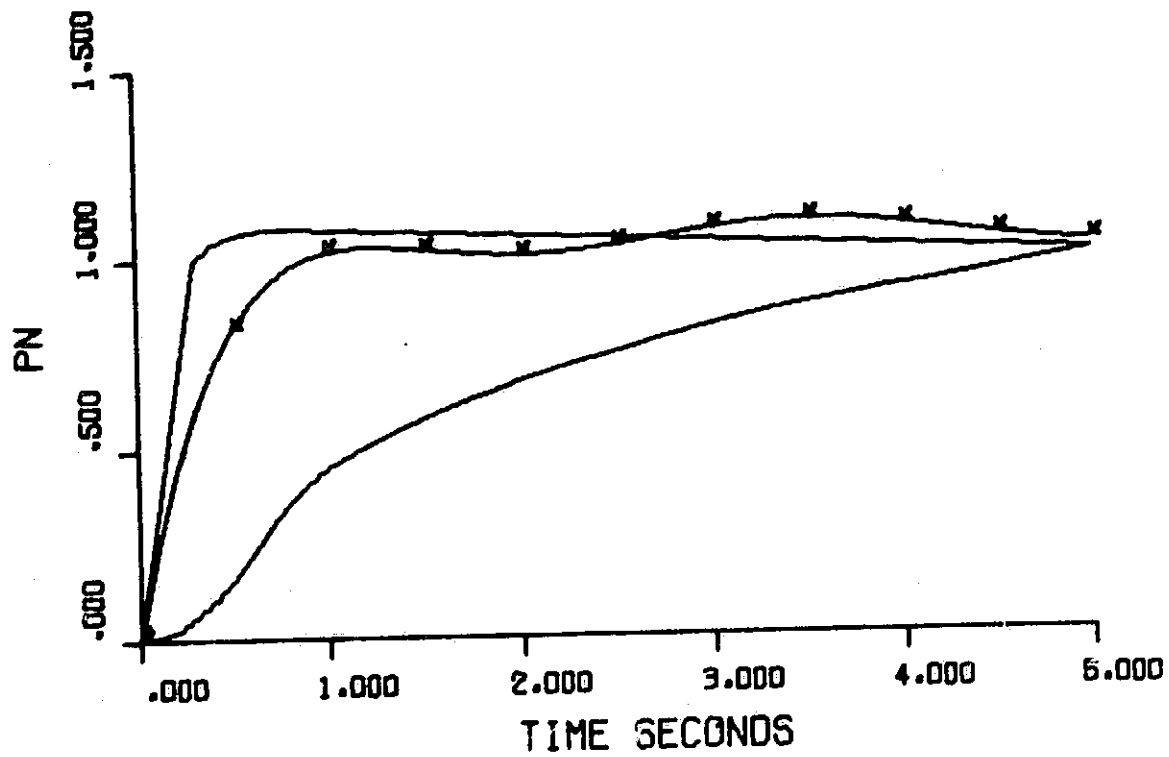
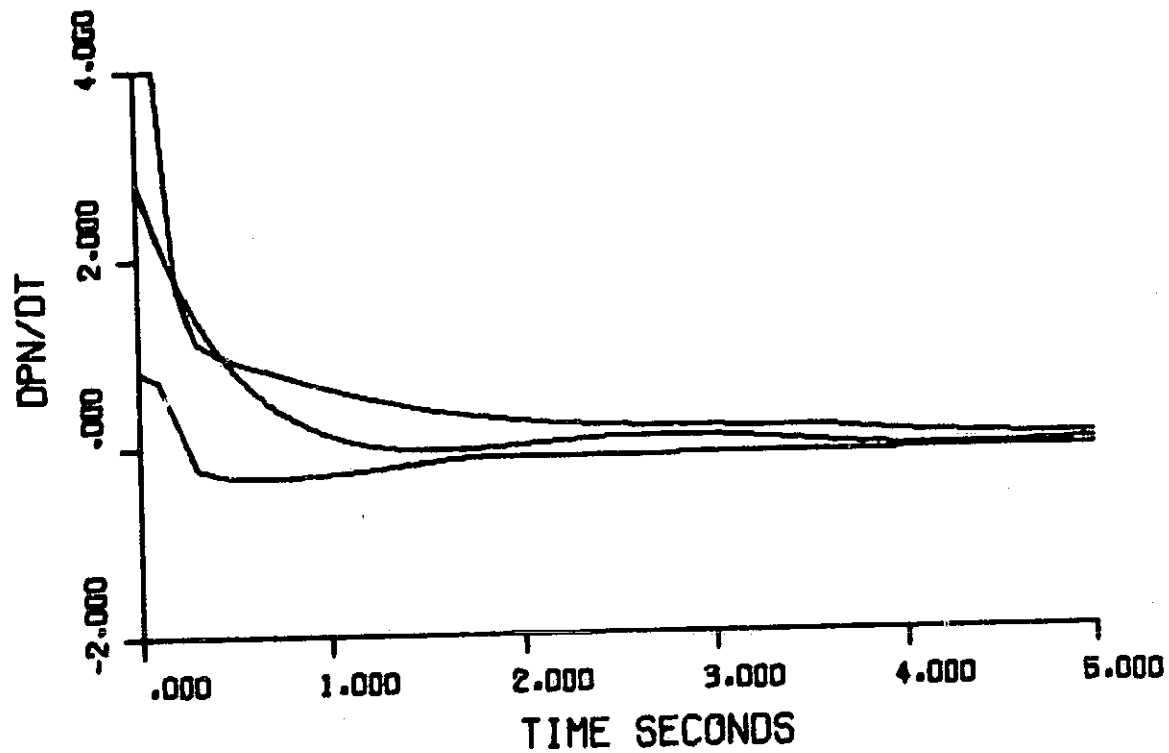


FIG.17 ROLLRATE

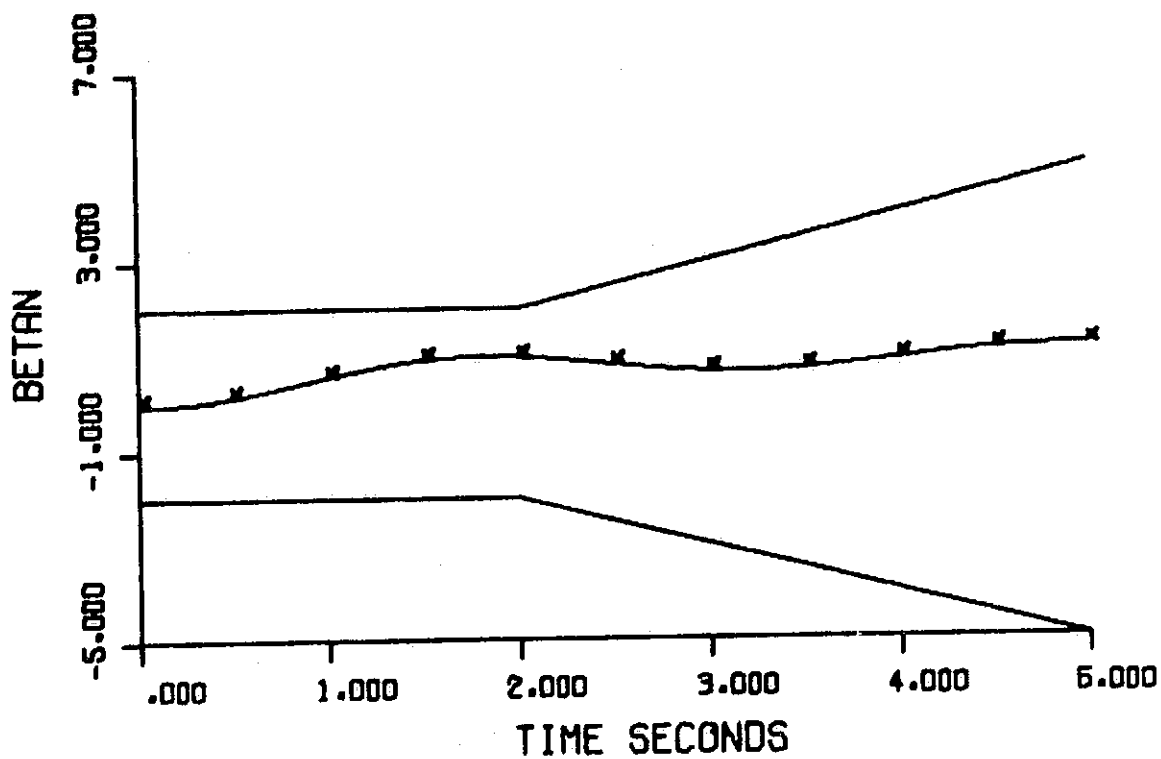
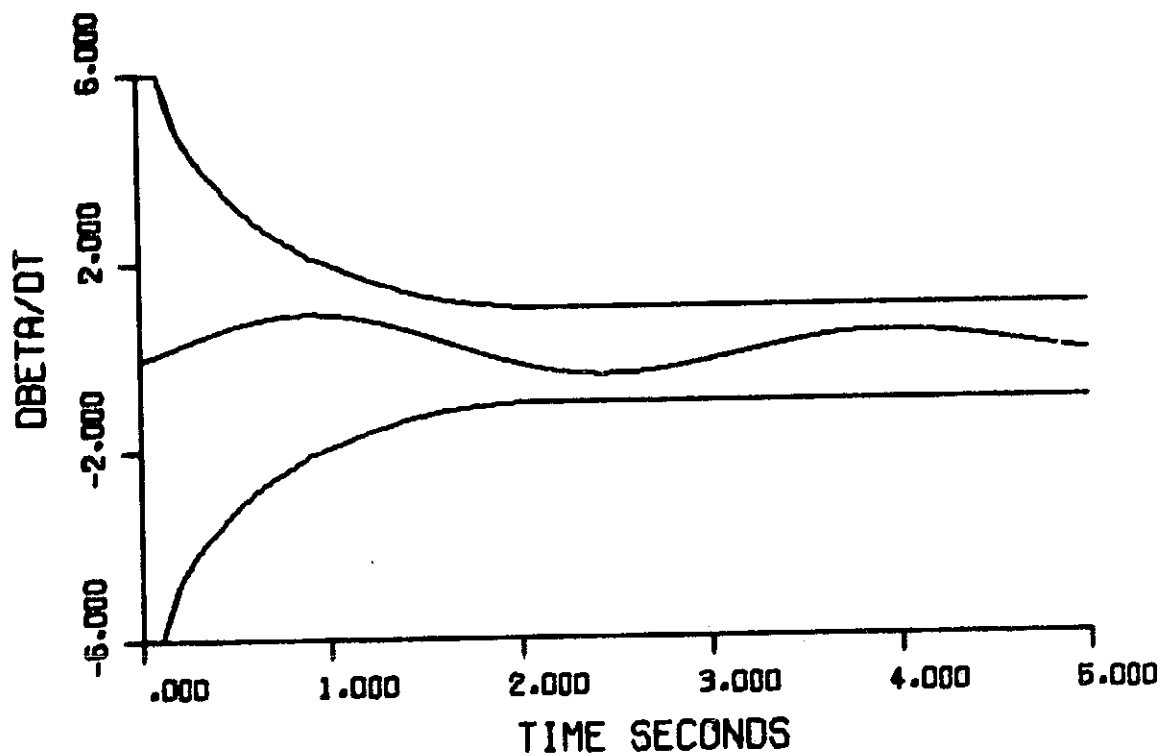


FIG.18 SIDESLIP

ORIGINAL PAGE IS
OF POOR QUALITY

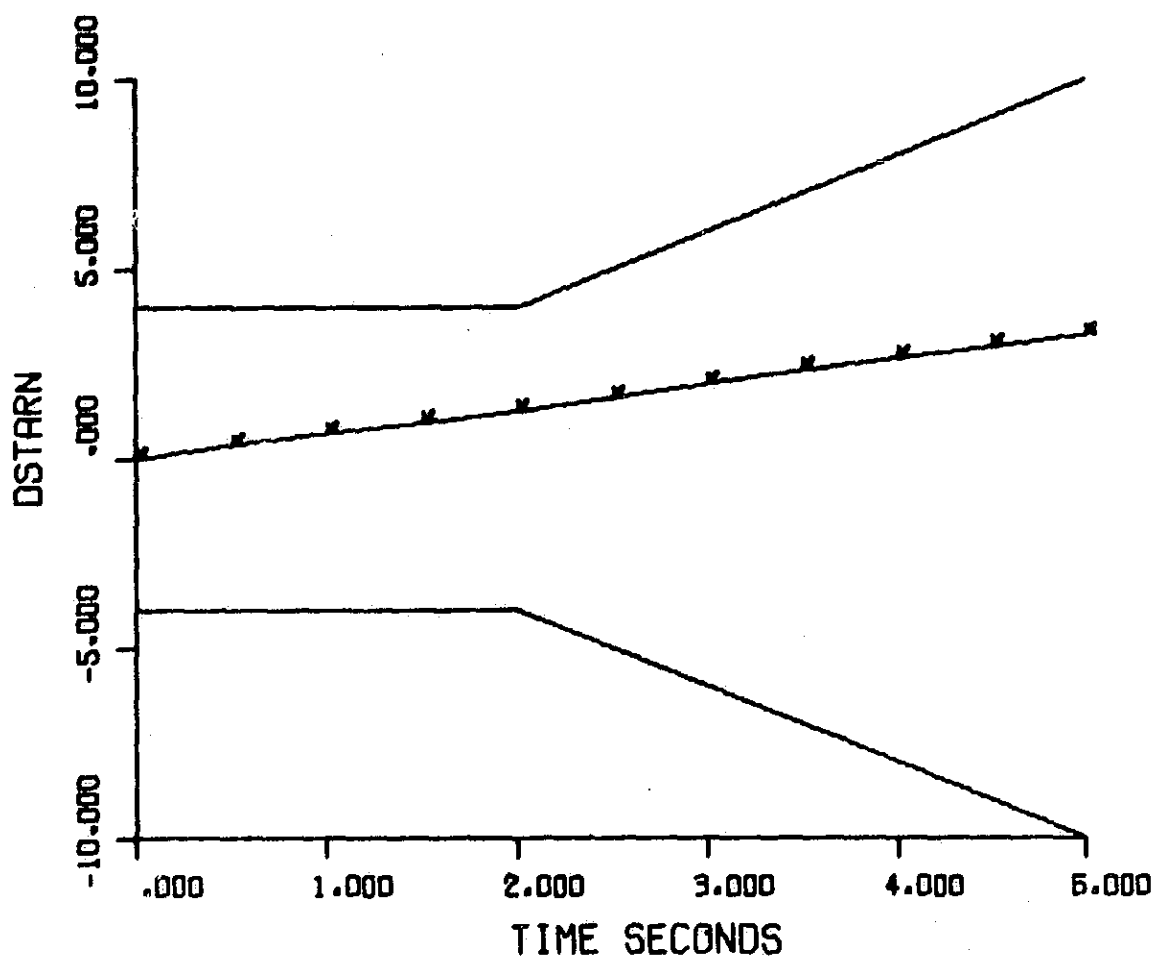


FIG.19A LATERAL CRITERION

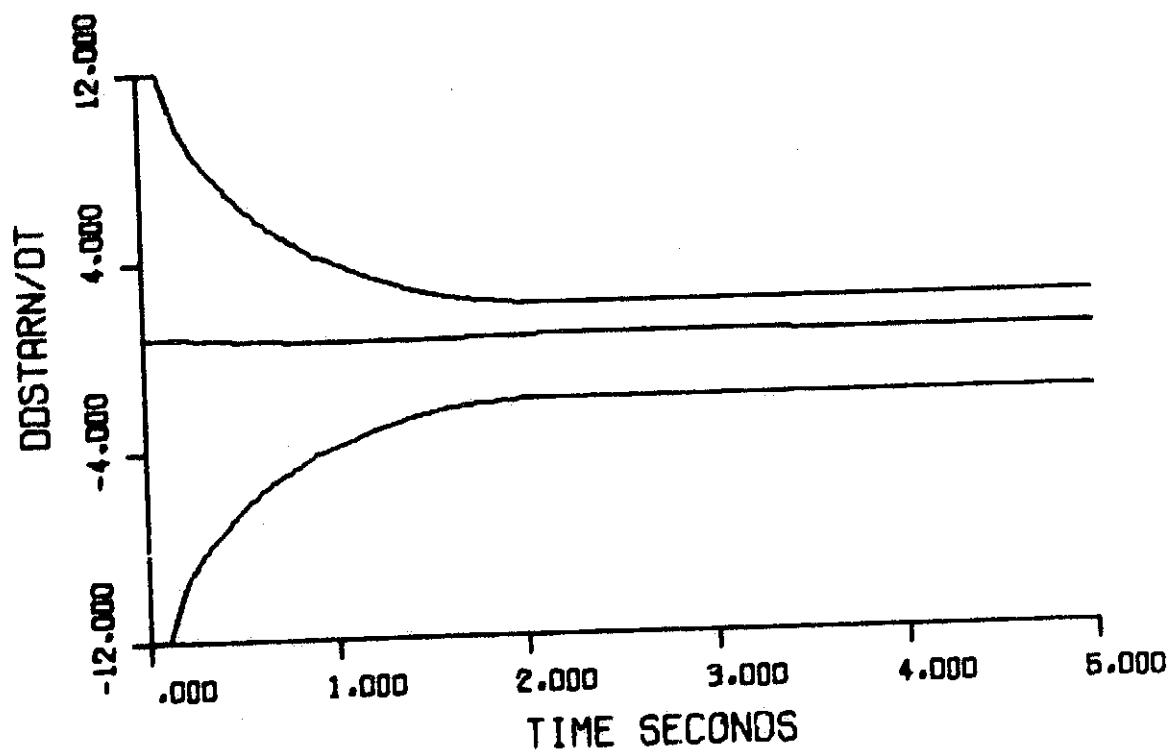


FIG. 19B

ORIGINAL PAGE IS
OF POOR QUALITY

$$\underline{b} = \begin{bmatrix} 5.818 \\ -.053 \\ -.215 \\ 0.050 \end{bmatrix}$$

$$\underline{G} = \begin{bmatrix} 1.0 & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \\ 14.35 & -11.13 & -143.5 & 32.61 \end{bmatrix}$$

$$\underline{\tilde{h}} = \begin{bmatrix} 0. \\ 0. \\ 0. \\ -.023 \end{bmatrix}$$

These results are close to the original arrays. The differences are attributed to the truncation of the CONTROL output values to two or three significant figures to approximate imprecise or sketched input data.

DISCUSSION

The problem of obtaining an $\underline{A}$ and $\underline{b}$ from specified output time histories is one of nonlinear, noise-free identification. Five techniques for solving this problem have been suggested. The minimization of a cost functional which measures the differences between a trial solution and the handling-quality time-history envelopes would consume a large amount of computer time and there is no assurance that such a cost functional would be sufficiently well behaved to have a useful solution. Similarly, one could consume a large amount of computer time seeking solutions by random direct search. A graduate student is currently working on a variation of direct search in which the sensitivity of the time-history errors to changes in the $\underline{A}$ matrix elements is calculated. Then a set of incremental changes to the $\underline{A}$ elements can be obtained.

The Laplace transformation method is also being pursued by a graduate student. In this form the problem is quite easily formulated but is ill-posed. It remains to be seen whether this method can advantageously incorporate the loose bounds on the eigenvalues or not. It also has the disadvantage of being a two-approximate-step process. One obtains $\underline{y}(t)$ from $\hat{\underline{y}}(t)$ and then $\tilde{\underline{y}}(t)$ from $\underline{y}(t)$ except that the latter two will not coincide as they do in the pseudodata method. The Laplace transformation method and the sensitivity matrix method do have the advantage that they can be made to yield $\underline{A}$ matrices of the form of equation 19.

The remaining approach, the pseudodata method has two advantages. It contains only one approximation step and it is numerically efficient. The disadvantages are that it is somewhat less general and yields unconstrained $\underline{A}$ matrices. Experience has shown that the transfer functions resulting from

the unconstrained $\underline{A}$ matrices resemble those produced by $\underline{A}$ matrices of the form of equation 19. The pseudodata method results in a well-posed set of bilinear algebraic equations which yield an $\underline{A}$ matrix having the specified eigenvalues.

The pseudodata method is the only method which readily achieves zero suppression. In the other methods zero suppression contributes to their ill-posedness making useful solutions even more numerically difficult to obtain. The utility of any zero-suppressed solution is called into doubt by the detrimental effect suppression has on the LSE fit of the discretized input data (see Figures 13, 14, 15, and 16). Relieved of the need to suppress transfer-function zeros, one might prefer one of the other methods.

APPENDIX A
Use of Program AANDB

INTENDED OPERATION

1. Put in normalized discrete data
2. Obtain normalized responses
3. Put in normalization factors
4. Obtain the normalizing distribution matrix
5. Obtain the non-normalized airplane equations in state-space form.
6. Use CONTROL to obtain transfer functions

Steps 1 through 5 are performed by an example Fortran program AANDB for the fourth-order small-lateral-motion case. The structure of the program is shown in Figure A-1. The subroutines perform the following tasks:

MUGEN: Calculates eigenvalues from input time-history data.

CASE1* (with entry points CASE2 and CASE3): Calculates the c_i required to fit $c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t} + c_3 e^{\lambda_3 t} + c_4 e^{\lambda_4 t} + c_5$ to the input time-history data.

LISTER: Prints information from COMMON on demand.

FITTING: Fits a polynomial to roll-rate data and integrates it to produce roll-angle pseudodata.

MODEL: Calculates the plant matrix, $\underline{A}$, input distribution vector, $\underline{b}$, the output distribution matrix, $\underline{G}$, and the input/output coupling vector, $\underline{h}$.

RESPONS: Integrates the plant equations to produce comparison time histories.

* CASE1 does not suppress any transfer function zeros, CASE2 suppresses one zero and CASE 3 suppresses two zeros.

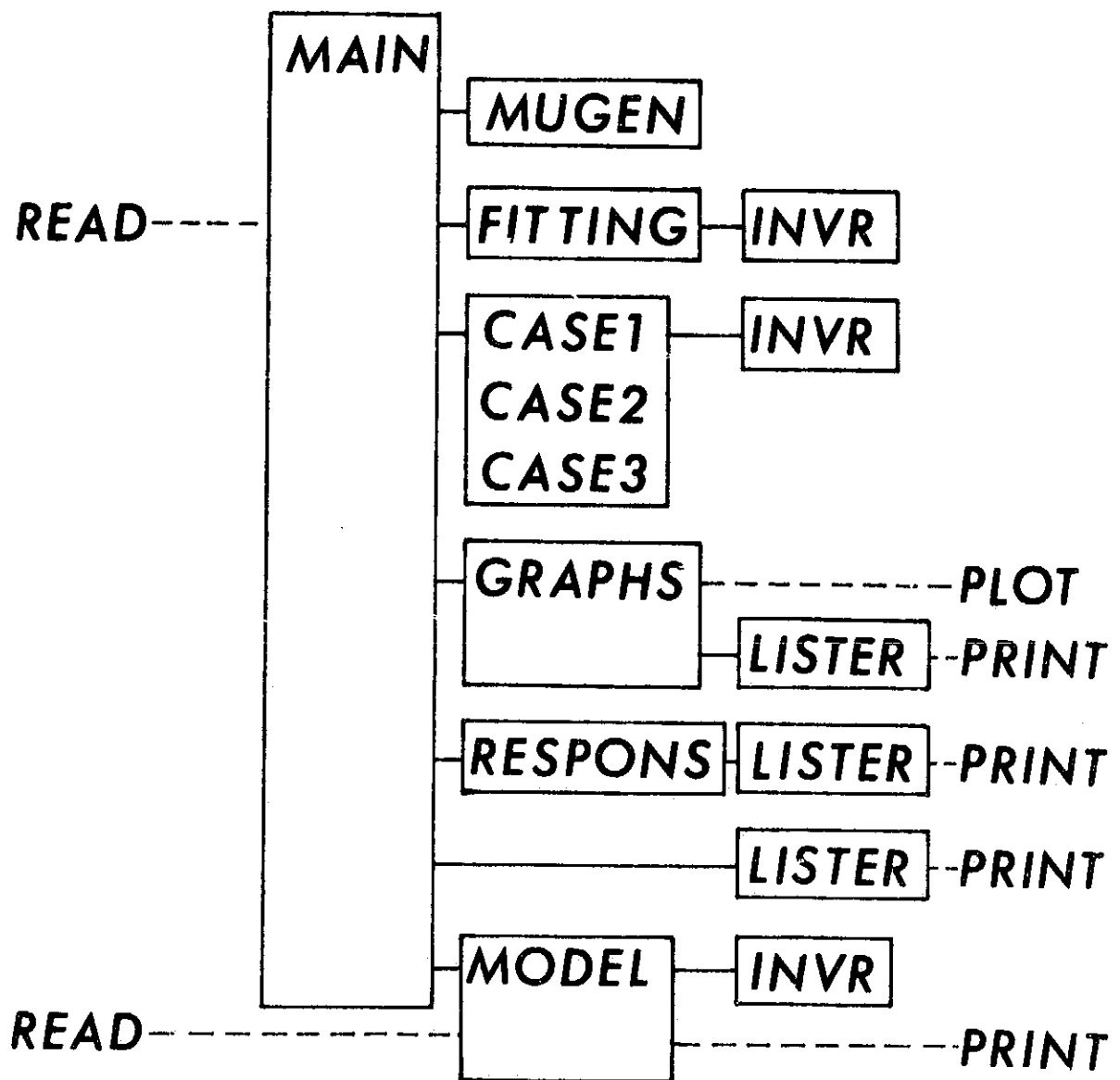


FIG. A1

ORIGINAL PAGE IS
OF POOR QUALITY

GRAPHS: Plots envelopes and time histories and their first derivatives.

INVR: Inverts matrices.

The first fourteen input data cards are read by MAIN. The final two input data cards are read by subroutine MODEL. The input data cards contain the following information:

1st data card: NN(20); 2012

IF NN(1) = 1: LISTER will print: data, NN, MCASE, NPLOT, NHIST,
input time histories

NN(2) = 1: LISTER will print eigenvalues obtained from input
time histories

NN(3) = 1: LISTER will print eigenvalues specified by user

NN(4) = 1: LISTER will print roll angle data obtained from
roll rate

NN(5) = 1: LISTER will print complex coefficient matrix $\underline{C}$

NN(6) = 1: LISTER will print summary of time-history curve fitting results

NN(7) = 1: not used

NN(8) = 1: not used

NN(9) = 1: LISTER will print envelopes for PN, BETAN, DSTAR
& derivatives

NN(10) = 1: LISTER will print fitted curves for PN, BETAN, DSTAR
& derivatives

NN(11) = 1: LISTER will print number of terms included in series
for $e^{\underline{A}\Delta t}$

NN(12) = 1: LISTER will print difference equation $\underline{P}$ matrix

NN(13) = 1: LISTER will print difference equation $\underline{g}$ vector

NN(14) = 1: LISTER will print responses obtained by integrating
model equations

NN(15) = 1: LISTER will print derivatives of responses obtained
from $\underline{P}$, $\underline{g}$

NN(16) = 1: LISTER will print "PN PLOTTED" if NPLLOT requests it
 NN(17) = 1: LISTER will print "BETAN PLOTTED" if NPLLOT requests it
 NN(18) = 1: LISTER will print "DSTAR PLOTTED" if NPLLOT requests it
 NN(19) = 1: not used
 NN(20) = 1: not used

2nd - 12th data cards: TIME, PN, BETAN, DSTAR; 4F10.0

0.0	0.0	0.0	0.0
0.5	PN(.5)	BETAN(.5)	DSTAR(.5)
1.0	PN(1.)	BETAN(1.)	DSTAR(1.)
1.5	PN(1.5)	BETAN(1.5)	DSTAR(1.5)
2.0	PN(2.)	BETAN(2.)	DSTAR(2.)
2.5	PN(2.5)	BETAN(2.5)	DSTAR(2.5)
3.0	PN(3.)	BETAN(3.)	DSTAR(3.)
3.5	PN(3.5)	BETAN(3.5)	DSTAR(3.5)
4.0	PN(4.)	BETAN(4.)	DSTAR(4.)
4.5	PN(4.5)	BETAN(4.5)	DSTAR(4.5)
5.0	PN(5.)	BETAN(5.)	DSTAR(5.)

13th data card: MCASE, NPLLOT, NHIST; 3I4

If NHIST $\neq$ 0 subroutine RESPON will be called to calculate the
 model time histories. If NPLLOT $\neq$ 0 subroutine graphs will be called
 to plot the model time histories according to:

ORIGINAL PAGE IS
 OF POOR QUALITY

NPLOT

- 0 no plots
- 1 PN and PNDOT plotted
- 2 BETAN and BETANDOT plotted
- 3 PN, PNDOT, BETAN, BETANDOT plotted
- 4 PSTAR, DSTARDOT plotted
- 5 BETAN, BETANDOT, DSTAR, DSTARDOT plotted
- 6 PN, PNDOT, DSTAR, DSTARDOT plotted
- 7 all plots
- 8 no plots

MCASE must be appropriate for:

GO TO(1,2,3,4,5,6,7,8,9), MCASE

- 1 $P(s)$ has no zero(s) suppressed
 $\beta(s)$ has no zero(s) suppressed
- 2 $P(s)$ has two zero(s) suppressed
 $\beta(s)$ has one zero(s) suppressed
- 3 $P(s)$ has one zero(s) suppressed
 $\beta(s)$ has two zero(s) suppressed
- 4 $P(s)$ has one zero(s) suppressed
 $\beta(s)$ has one zero(s) suppressed
- 5 $P(s)$ has one zero(s) suppressed
 $\beta(s)$ has no zero(s) suppressed
- 6 $P(s)$ has no zero(s) suppressed
 $\beta(s)$ has one zero(s) suppressed
- 7 $P(s)$ has two zero(s) suppressed
 $\beta(s)$ has no zero(s) suppressed
- 8 $P(s)$ has no zero(s) suppressed
 $\beta(s)$ has two zero(s) suppressed
- 9 $P(s)$ has two zero(s) suppressed
 $\beta(s)$ has two zero(s) suppressed

14th data card: specified eigenvalues; 8F10.0

1-10 First eigenvalues which is real
21-30 Second eigenvalue which is also real
41-50 Real part of third eigenvalue
51-60 Imaginary part of third eigenvalue
61-70 Real part of fourth eigenvalue
71-80 Imaginary part of fourth eigenvalue

15th card: velocity, length, c_3 , q_{co} , p_{ss} , β_{ss} , D_{ss}^* ; 7F10.0

VELOCITY: Nominal airspeed in ft/sec

LENGTH: Centerline length from CG to pilot in ft

c_3 : Dimensional constant for D^*

q_{co} : Nominal dynamic pressure in lb/ft²

p_{ss} : Roll rate normalization factor

β_{ss} : Sideslip normalization factor

D_{ss}^* : DSTAR normalization factor

16th data card: Newton-Euler parameters; 14, F20.0

ITMAX, 14, Maximum number of iterations of Newton-Euler algorithm

EPSI, F20.0, when every unknown (the elements of the A matrix) changes by an amount smaller than EPSI, the Newton-Euler Algorithm stops

Note: 50,0.00001 seem to work well.

INTERPRETATION OF THE PRINTED OUTPUT

If NN = 20*1 all of the following output will be produced:

ORIGINAL PAGE IS
OF POOR QUALITY

LISTER calls from the main program print:

1. Date, list table NN, case number, plot request code, time response code and the input time history data.

2. Real and imaginary parts of each MU(4) and EI(4) obtained from the input time histories. The EI are eigenvalues, λ_i , and each MU is $\mu_i = e^{0.5\lambda_i}$ where 0.5 is the uniform time interval between input time-history data points.

3. Real and imaginary parts of the specified eigenvalues and associated MU values. These are the eigenvalues used in the least-square fitting process to obtain $\underline{C}$, not the quantities derived from the data. The derived values are presented only for comparison purposes.

4. The PHI or roll-rate pseudodata generated from the PN or roll rate input time history. The time intervals are the same as for the original data.

5. The TIME RESPONSE COEFFICIENTS matrix, $\underline{C}'$, is printed. This is a 4 x 5 array of complex numbers:

$$\underline{y} = \underline{C}e^{\underline{\Lambda}t} + \underline{c} \quad , \quad \text{where } \underline{C}' = [\underline{C}; \underline{c}] \quad ,$$

which analytically represents the input time histories. This $\underline{y}(t)$ is printed as: FITTED TIME RESPONSES by a call to LISTER from RESPON.

6. A summary printout of MU, EI, $\underline{C}$ and $\underline{c}$

LISTER calls from subroutine RESPON print:

1. The envelopes, upper and lower boundaries, for $p_n \beta_n$, D^* , $\frac{d}{dt} p_n$, $\frac{d}{dt} \beta_n$, and $\frac{d}{dt} D^*$. Values are given for every 0.1 seconds. These values are stored

in DATA statements in the main program.

2. The curves which have been fitted to the input time-history data are printed for p_n , β_n and D^* . The analytic expressions for the curves are differentiated and tabulated also giving $\frac{d}{dt} p_n$, $\frac{d}{dt} \beta_n$ and $\frac{d}{dt} D^*$.

3. The number of terms taken to calculate the truncated series used to represent $e^{-A\Delta t}$ is PAYNTERS RECIPE NUMBER. See CONTROL, Takahashi, et. al., page 103 [6].

4. The difference equation parameters $\underline{p}$ and $\underline{q}$ in:

$$\underline{x}_{k+1} = \underline{p}\underline{x}_k + \underline{q}u_k, \quad \underline{x}_0 = \begin{bmatrix} 0 \\ D_{\delta a} / D_r \\ 0 \\ 0 \end{bmatrix}$$

5. The numerically integrated response time histories $\underline{y}_k$ where $\underline{y}_k = \underline{G}\underline{x}_k + \underline{h}u_k$.

6. The first derivatives of the integrated responses are tabulated at 0.1 second intervals. They are obtained from

$$\dot{\underline{x}}_k = \underline{A}\underline{x}_k + \underline{b}u_k$$

and

$$\dot{\underline{y}}_k = \underline{G}\dot{\underline{x}}_k$$

LISTER calls from subroutine GRAPHS prints:

1. If NPLOT is such that plots are requested, LISTER will print "PN

PLOTTED" if p_n and $\frac{d}{dt} p_n$ plots have been generated, "BETAN PLOTTED" if β_n and $\frac{d}{dt} \beta_n$ plots have been generated and "DSTAR PLOTTED" if D^* and $\frac{d}{dt} D^*$ plots have been generated. The envelopes are automatically added to each plot.

SAMPLE PLOTTED OUTPUT

Plots are produced in three groups which can be requested individually or in any combination. These groups, the PN group, the BETAN group and the DSTAR group, consist of a title line, a lower graph and an upper graph. Each lower graph shows the upper envelope boundary, the lower envelope boundary, the analytical curve which has been fitted to the input data and the time history obtained by integrating the equations of motion. These four traces are represented by continuous lines and are plotted versus time on the horizontal scale. The fitted and integrated lines should coincide. In addition, the lower graph contains eleven discrete symbols representing the input data.

Each upper graph shows the upper envelope boundary, the lower envelope boundary, the first derivative of the fitted analytical curve and the first derivative of the integrated time response. The last two should coincide. All four curves are represented by continuous lines and are plotted versus time on the horizontal scale. A PN group sample is shown in Figure A-2, a BETAN group sample is shown in Figure A-3 and a DSTAR group is shown in Figure A-4. The example plots are unusual in that the original data was generated by a linear simulation program using a fourth-order model. Thus one should expect excellent agreement between the input data and the fitted and integrated results. Data obtained from other sources will not be matched as well.

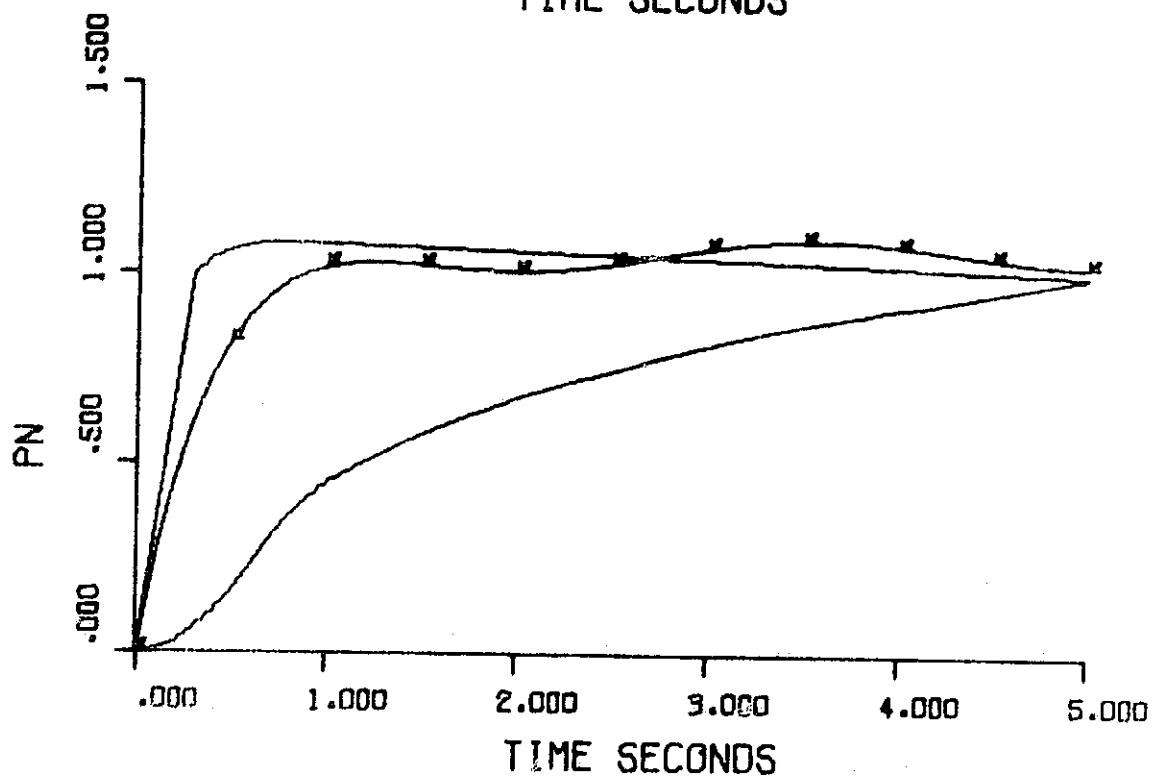
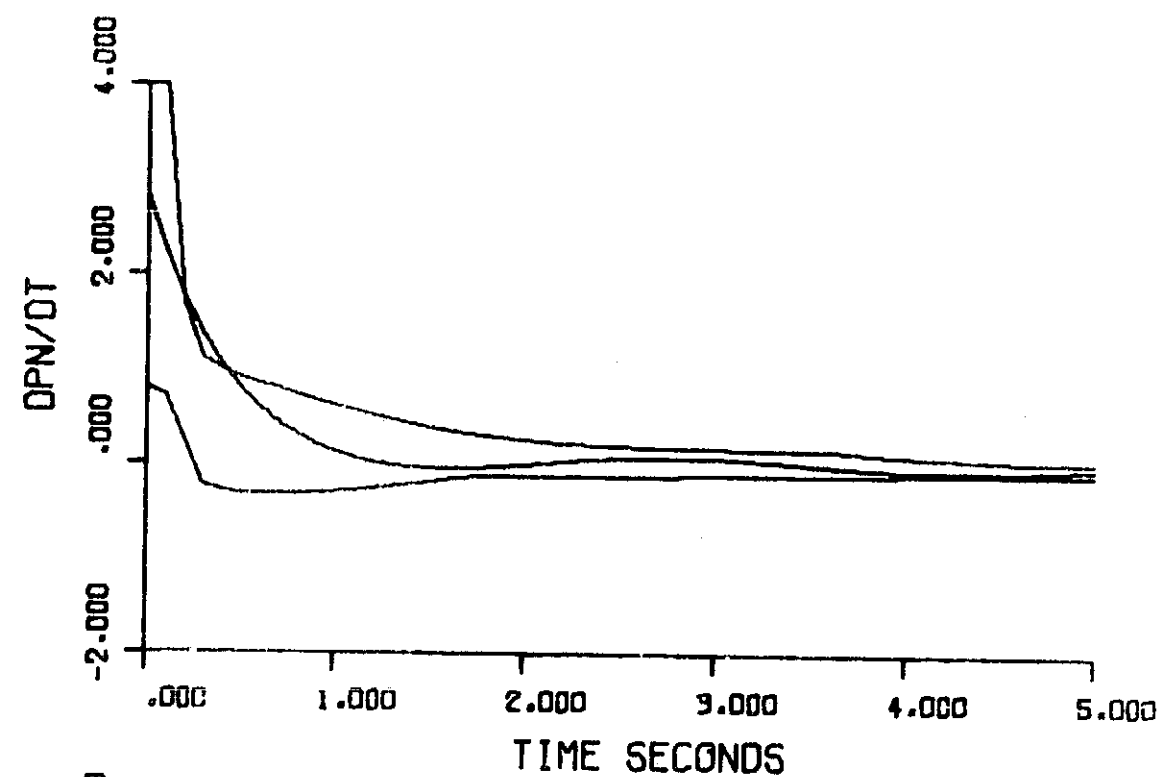


FIG. A2 ROLL RATE

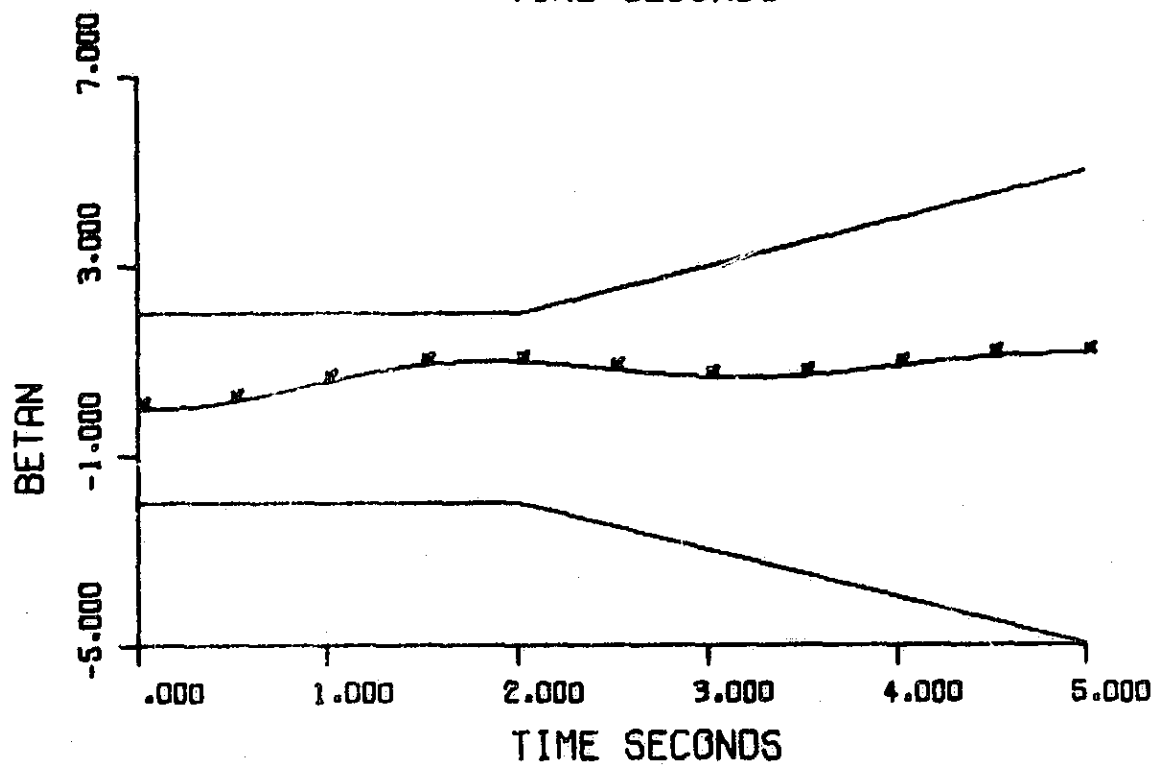
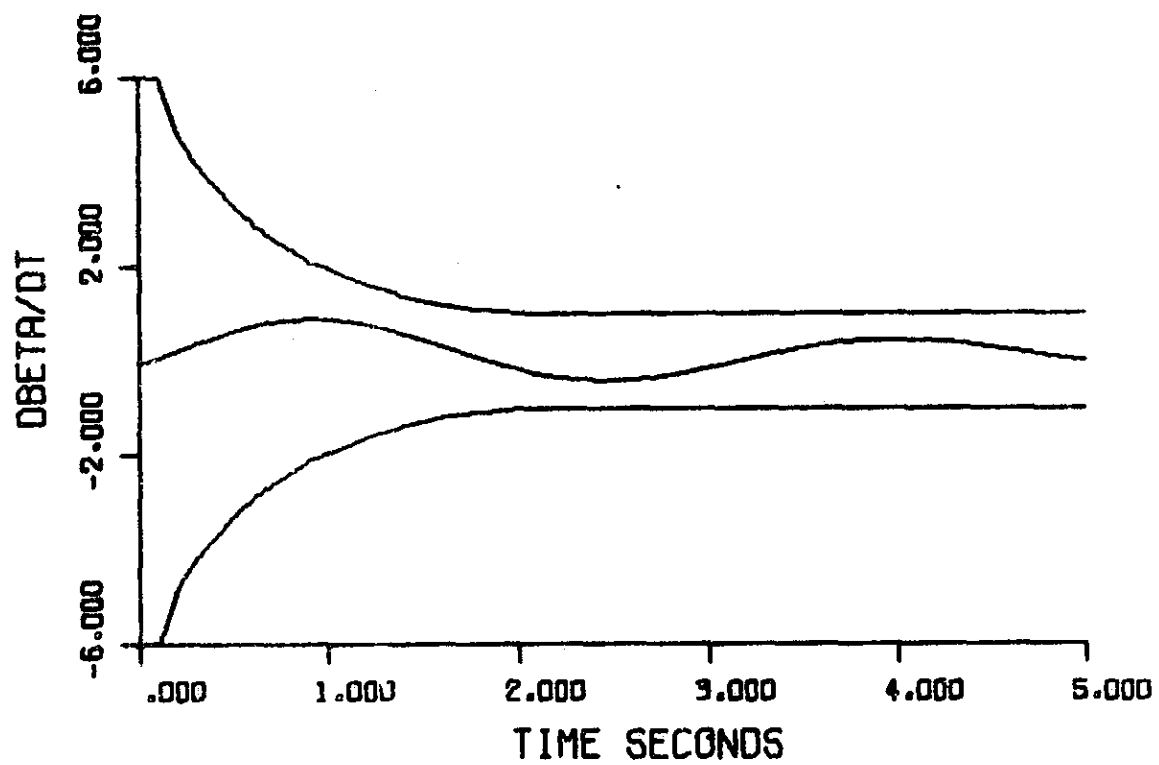


FIG. A3 SIDESLIP

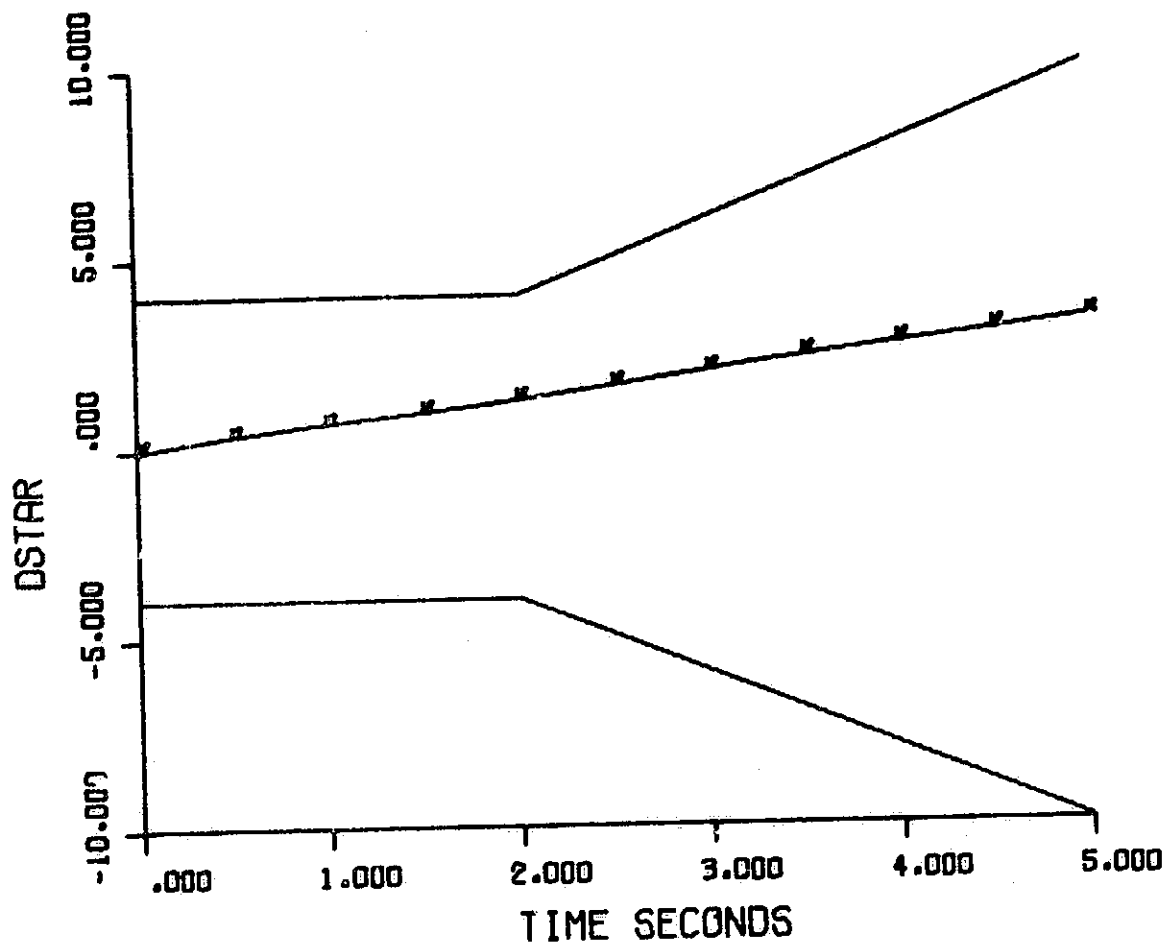


FIG. A4A

LATL. CRIT.

ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE A-4

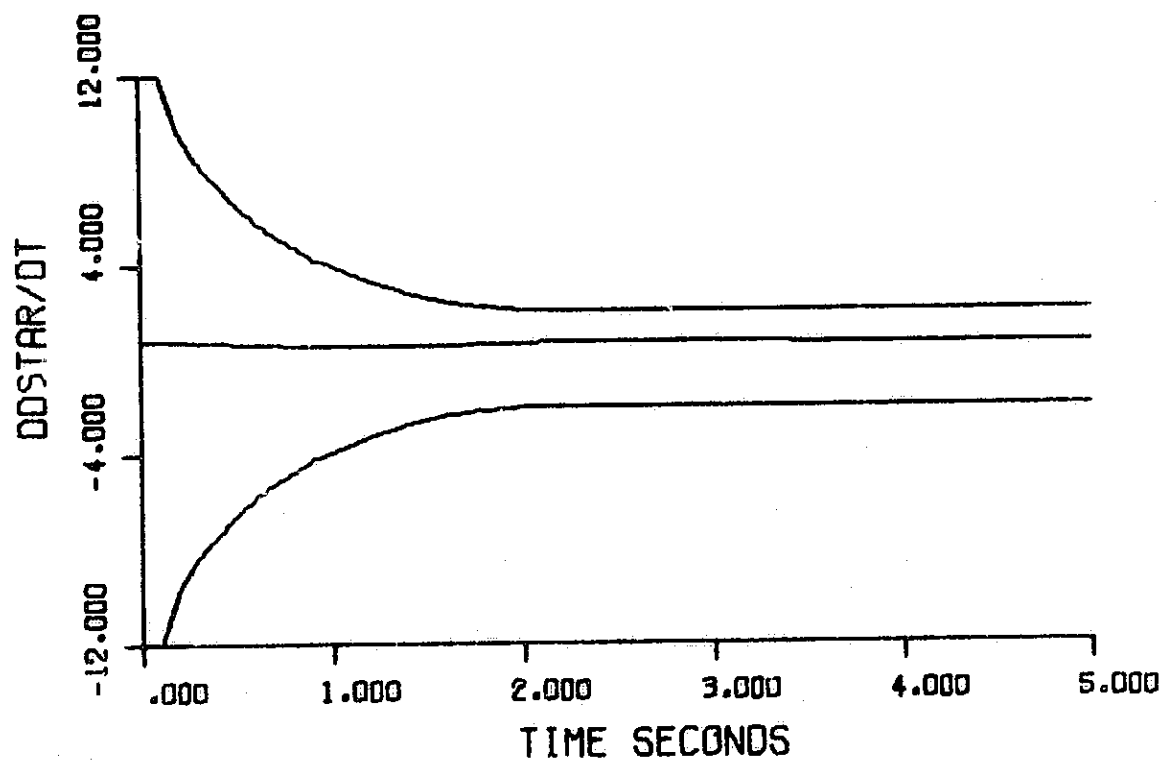


FIG. A4B

EXPERIENCE WITH THE PROGRAM

Several bounds on program flexibility exist. The first involves the generation of eigenvalues from input-time history data. Subroutine MUGEN does calculate a set of λ_i from each set of eleven points representing one input time history. These values are not correct or consistent even when the data is carefully contrived for best results. This is due to the dependence of the λ_i calculation upon differences between adjacent (in time) data values. In anticipation of the program being normally used with data obtained graphically, the values used in the example were held to three significant figures. This so degraded the λ_i calculation that the values from the three input time histories varied from the known values used to generate the data by two orders of magnitude in the case of small eigenvalues and occasionally had the opposite sign. In general, the complex eigenvalues were more closely identified than the real eigenvalues. This was particularly true if the data precision was increased. In view of the anticipated character of the input data and the desirability of specifying the eigenvalues of the resulting model, no further development of MUGEN is planned.

A second difficulty arises when zero suppression is specified. Subroutine CASE2 and CASE3 calculate coefficients, C , which yield transfer functions having the correct pole-zero excesses. The resulting expression will normally be a poor representation of the input time-history data but that is unavoidable. Unfortunately, the suppression of zeros also tends to ill-condition the numerical equations which must be solved to obtain A . In extreme cases this prevents convergence of the simple Newton-Euler-Raphson algorithm employed in MODEL.

ORIGINAL PAGE IS
OF POOR QUALITY

Finally, there is a general difficulty that may interfere with the integration of the model equations of motion in RESPONS. If the $\underline{A}$ matrix is ill-conditioned an unworkable number of terms in the series approximation for $\underline{e}^{\underline{A}\Delta t}$ may be required. This is caused by the fixed Δt of 0.1 seconds which is built into RESPONS. A limit of 30 terms is imposed. Other limitations will undoubtedly come to light as experience with the program accumulates.

APPENDIX B
Program AANDB Listing

```

PROGRAM AANDB(INPUT, OUTPUT)
COMMON /NAMES/NN(20),PN(11),BETAN(11),DSTAR(11),TPN(11)
1      ,PNF(51),BETANF(51),DSTARF(51),PN1F(51),BETAN1F(51),DSTAR1F(51)
2AR1F(51),PNC(51),BETANC(51),DSTARC(51),PN1C(51),BETAN1C(51),DSTAR1C(51)
3C(51),APN(11),ABETAN(11),ADSTAR(11),BPN(11),BBETAN(11),BDSTAR(11),MAI
4PHIN(11)
COMMON /PARAM/MU,EIGEN,      C,NCASE,MCASE,NZ,      NPLOT,
1NHIST,PNFINAL,BNFINAL,DSFINAL,TIME(51),P(4,4),Q(4)
COMMON /LIMITS/PNU(51),PNL(51),BETANU(51),BETANL(51),DSTARU(51),
1DSTARL(51),PN1U(51),PN1L(51),BETAN1U(51),BETAN1L(51),DSTAR1U(51),
2DSTAR1L(51)
COMMON /FINAL/ PSS,RETASS,DSTARSS,DP,DR,DBETA,DPHI,A(4,4),B(4),DDEMAI
1LTA
COMMON MU(4),EIGEN(4),C(5,4)
DATA PNU/.0, .333,.667,1.0,1.043,1.064,1.076,1.08,1.08,1.078,1.076,MAI
11.074,1.072,1.07,1.059,1.067,1.065,1.063,1.061,1.059,1.057,1.055,1MAI
2.053,1.051,1.05,1.048,1.045,1.044,1.042,1.04,1.038,1.036,1.034,1.0MAI
332,1.03,1.029,1.027,1.025,1.023,1.021,1.019,1.017,1.015,1.013,1.01MAI
41.01,1.008,1.006,1.004,1.002,1.0/PNL/0.0,0.01,0.027,0.065,0.107MAI
5.0.16,0.227,0.3,0.35,0.409,0.45,0.479,0.507,0.53,0.556,0.579,0.6,0.6MAI
62,0.639,0.656,0.677,0.692,0.708,0.722,0.733,0.748,0.762,0.778,0.79MAI
7.0.804,0.816,0.829,0.84,0.852,0.861,0.871,0.88,0.888,0.896,0.908,0MAI
8.915,0.92,0.929,0.938,0.947,0.956,0.965,0.974,0.983,0.992,1.0/
DATA BETANU/21*2.0,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.0,3.1,3.2MAI
1.3,3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.0,4.1,4.2,4.3,4.4,4.5,4.6,4.7,4.8,4MAI
2.9,5.0,BETANL/21*-2.0,-2.1,-2.2,-2.3,-2.4,-2.5,-2.6,-2.7,-2.8,-2MAI
39,-3.0,-3.1,-3.2,-3.3,-3.4,-3.5,-3.6,-3.7,-3.8,-3.9,-4.0,-4.1,-4.2MAI
4,-4.3,-4.4,-4.5,-4.6,-4.7,-4.8,-4.9,-5.0/
DATA DSTARU/21*4.0,4.2,4.4,4.5,4.8,5.0,5.2,5.4,5.6,5.8,6.0,6.2,6.4MAI
1.6,6.6,6.8,7.0,7.2,7.4,7.6,7.8,8.0,8.2,8.4,8.6,8.8,9.0,9.2,9.4,9.6,9.8MAI
2.8, 10.0/DSTARL/21*-4.0,-4.2,-4.4,-4.6,-4.8,-5.0,-5.2,-5.4,-5.6,-5.8MAI
36,-5.8,-6.0,-6.2,-6.4,-6.6,-6.8,-7.0,-7.2,-7.4,-7.6,-7.8,-8.0,-8.2MAI
4 2,-8.4,-8.6,-8.8,-9.0,-9.2,-9.4,-9.6,-9.8,-10.0/
DATA PN1U/4.0,4.0,1.686,1.122,0.992,0.908,0.943,0.797,0.724,0.663,MAI
10.613,0.556,0.510,0.464,0.421,0.379,0.349,0.318,0.295,0.272,0.257,MAI
20.234,0.222,0.211,0.199,0.192,0.188,0.184,0.180,0.176,0.172,0.168,MAI
30.165,0.169,0.165,0.165,0.157,0.142,0.119,0.107,0.096,0.080,0.073,MAI
40.069,0.061,0.054,0.046,0.046,0.046,0.045,0.044,0.044,0.044,0.044,MAI

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

50.255,-.224,-.297,-.323,-.327,-.323,-.319,-.304,-.289,-.274,-.251,MAI 390
6-.229,-.205,-.183,-.148,-.129,-.125,-.118,-.125,-.123,-.121,-.120,MAI 400
7-.119,-.116,-.114,-.112,-.110,-.109,-.107,-.105,-.103,-.102,-.100,MAI 410
8-.098,-.096,-.095,-.093,-.091,-.089,-.088,-.086,-.084,-.082,-.081,MAI 420
9-.079,-.077,-.075,-.074,-.072/MAI 430
DATA BETAN1/6.0,6.0,4.80,4.15,3.72,3.28,2.91,2.61,2.37,2.10,1.95,MAI 440
11.78,1.61,1.49,1.35,1.26,1.17,1.12,1.07,1.05,1.00,30*1./,RETAN1L/MAI 450
2-6.0,-6.0,-4.80,-4.15,-3.72,-3.28,-2.91,-2.61,-2.37,-2.10,-1.95,MAI 460
3-1.78,-1.61,-1.49,-1.35,-1.26,-1.17,-1.12,-1.07,-1.05,-1.00,30*-1. MAI 470
40/MAI 480
DATA DSTAR1U/12.0,12.0,9.65,8.30,7.44,6.56,5.82,5.22,4.74,4.20,MAI 490
13.90,3.55,3.22,2.98,2.70,2.52,2.34,2.24,2.14,2.10,2.00,30*2.0/,MAI 500
2DSTAR1L/-12.0,-12.0,-9.60,-8.30,-7.44,-6.56,-5.82,-5.22,-4.74,MAI 510
3-4.20,-3.90,-3.56,-3.22,-2.98,-2.70,-2.52,-2.34,-2.24,-2.14,-2.10,MAI 520
4-2.00,30*-2.0/MAI 530
READ 300,NNMAI 540
READ 100,(TPN(I),PN(I),BETAN(I),DSTAR(I),I=1,11)MAI 550
READ 200,MCASE,NPLOT,NHISTMAI 560
IF(NN(1).EQ.1) CALL LISTER(1)MAI 570
CALL MUGEN(PN,MU,EIGEN)MAI 580
IF(NN(2).EQ.1) CALL LISTER(2)MAI 590
CALL MUGEN(BETAN,MU,EIGEN)MAI 600
IF(NN(2).EQ.1) CALL LISTER(2)MAI 610
CALL MUGEN(DSTAR,MU,EIGEN)MAI 620
IF(NN(2).EQ.1) CALL LISTER(2)MAI 630
READ 400,EIGENMAI 640
MU(1)=CEXP(EIGEN(1)*.5)MAI 650
MU(2)=CEXP(EIGEN(2)*.5)MAI 660
MU(3)=CEXP(EIGEN(3)*.5)MAI 670
MU(4)=CONJG(MU(3))MAI 680
IF(NN(3).EQ.1) CALL LISTER(3)MAI 690
CALL FITTING(PN,TPN,APN)MAI 700
DO 10 I=1,11MAI 710
PHIN(I)=0. MAI 720
DO 10 J=1,11MAI 730
PHIN(I)=PHIN(I)+APN(J)* TPN(J)**(12-J)/FLOAT(12-J)MAI 740
IF(NN(4).EQ.1) CALL LISTER(4)MAI 750
GO TO (1,2,3,4,5,6,7,8,9),MCASEMAI 760
1 CALL CASE1(MU,PN,EIGEN,C(1,1))MAI 770
CALL CASE1(MU,BETAN,EIGEN,C(1,2))MAI 780

```

```

CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

2 CALL CASE3(MU,PN,EIGEN, C(1,1))
CALL CASE2(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

3 CALL CASE2(MU,PN,EIGEN, C(1,1))
CALL CASE3(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

4 CALL CASE2(MU,PN,EIGEN, C(1,1))
CALL CASE2(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

5 CALL CASE2(MU,PN,EIGEN, C(1,1))
CALL CASE1(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

6 CALL CASE1(MU,PN,EIGEN, C(1,1))
CALL CASE2(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

7 CALL CASE3(MU,PN,EIGEN, C(1,1))
CALL CASE1(MU,BETAN,EIGEN, C(1,2))
CALL CASE1(MU,PHIN,EIGEN, C(1,3))
CALL CASE1(MU,DSTAR,EIGEN, C(1,4))
IF(NN(5).EQ.1) CALL LISTER(5)
GO TO 11

```

```

MAY 790
MAY 800
MAY 810
MAY 820
MAY 830
MAY 840
MAY 850
MAY 860
MAY 870
MAY 880
MAY 890
MAY 900
MAY 910
MAY 920
MAY 930
MAY 940
MAY 950
MAY 960
MAY 970
MAY 980
MAY 990
MAY 1000
MAY 1010
MAY 1020
MAY 1030
MAY 1040
MAY 1050
MAY 1060
MAY 1070
MAY 1080
MAY 1090
MAY 1100
MAY 1110
MAY 1120
MAY 1130
MAY 1140
MAY 1150
MAY 1160
MAY 1170
MAY 1180

```

```

      8 CALL CASE1(MU,PN,EIGEN, C(1,1))
        CALL CASE3(MU,BETAN,EIGEN, C(1,2))
        CALL CASE1(MU,PHIN,EIGEN, C(1,3))
        CALL CASE1(MU,OSTAR,EIGEN, C(1,4))
      9 CALL CASE3(MU,PN,EIGEN, C(1,1))
        CALL CASE3(MU,BETAN,EIGEN, C(1,2))
        CALL CASE1(MU,PHIN,EIGEN, C(1,3))
        CALL CASE1(MU,OSTAR,EIGEN, C(1,4))
        IF(NN(5).EQ.1) CALL LISTER(5)
      11 IF(NN(5).EQ.1) CALL LISTER(6)
        CALL MCOEL(C,EIGEN,PSS,BETASS,OSTARSS,OP,OR,OBETA,DPHI,A,B,ODELTA)
        IF(NHIST.NE.0) CALL RESPON
        IF(NPLOT.NE.0) CALL GRAPHS
        STOP
      100 FORMAT(/F10.0)
      200 FORMAT(3I4)
      300 FORMAT(20I2)
      400 FORMAT(8F10.0)
        END

```

```

MAI 1190
MAI 1200
MAI 1210
MAI 1220
MAI 1230
MAI 1240
MAI 1250
MAI 1260
MAI 1270
MAI 1280
MAI 1290
MAI 1300
MAI 1310
MAI 1320
MAI 1330
MAI 1340
MAI 1350
MAI 1360
MAI 1370

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

SUBROUTINE MUGEN(D,MU,EIGEN)
DIMENSION O(11),PI(4,4),Q(4),PI(4,4),A(4)
COMPLEX MU(4),EIGEN(4),SAVE
P(1,4)=D(2)-D(1)
P(1,3)=D(3)-D(2)
P(1,2)=P(2,4)=D(4)-D(3)
P(1,1)=P(2,3)=D(5)-D(4)
P(2,2)=P(3,4)=Q(1)=D(6)-D(5)
P(2,1)=P(3,3)=D(7)-D(6)
P(3,2)=P(4,4)=Q(2)=D(8)-D(7)
P(3,1)=P(4,3)=D(9)-D(8)
P(4,2)=Q(3)=D(10)-D(9)
P(4,1)=D(11)-D(10)
D12=D(9)+3.*(D(11)-D(10))
Q(4)=D12-D(11)
CALL INVR(P,PI,4,0,4)
DO 1 I=1,4
  A(I)=0.
DO 1 J=1,4
  1 A(I)=A(I)+PI(I,J)*Q(J)

```

QUADRATIC SYNTHETIC DIVISION
NUMERICAL CALCULS
W. E. MILNE PAGE 53

```

X=0.
Y=0.
A0=1.
A1=-A(1)
A2=-A(2)
A3=-A(3)
A4=-A(4)
R0=A0
R1=A1-X*B0
R2=A2-X*B1-Y*R0
R3=A3-X*B2-Y*B1
R4=A4-X*B3-Y*B2

```

10

C
C
C
C
C
C
C

MUG 10
MUG 20
MUG 30
MUG 40
MUG 50
MUG 60
MUG 70
MUG 80
MUG 90
MUG 100
MUG 110
MUG 120
MUG 130
MUG 140
MUG 150
MUG 160
MUG 170
MUG 180
MUG 190
MUG 200
MUG 210
MUG 220
MUG 230
MUG 240
MUG 250
MUG 260
MUG 270
MUG 280
MUG 290
MUG 300
MUG 310
MUG 320
MUG 330
MUG 340
MUG 350
MUG 360
MUG 370
MUG 380
MUG 390

MUG	400
MUG	410
MUG	420
MUG	430
MUG	440
MUG	450
MUG	460
MUG	470
MUG	480
MUG	490
MUG	500
MUG	510
MUG	520
MUG	530
MUG	540
MUG	550
MUG	560
MUG	570
MUG	580
MUG	590
MUG	600
MUG	610
MUG	620
MUG	630
MUG	640
MUG	650
MUG	660
MUG	670
MUG	680
MUG	690
MUG	700
MUG	710
MUG	720
MUG	730
MUG	740
MUG	750
MUG	760
MUG	770
MUG	780

```

C0=80
C1=81-X*C0
C2=82-X*C1-V*C0
C3=  -X*C2-V*C1
D0=C2**2-C3*C1
DX=(93*C2-84*C1)/D0
DY=(84*C2-83*C3)/D0
X=X+DX
Y=Y+DY
IF(ABS(DX).GE.0.0001) GO TO 10
IF(ABS(DY).GE.0.0001) GO TO 10
U=81
V=82
ERROR=A4-B2*Y
IF(ABS(ERROR).LE.0.0001) GO TO 20
PRINT 100,ERROR
100 FORMAT(10X,6HERRCR=F10.7)
20 QUOT=X**2-4.*Y
IF(QUOT) 21,22,23
23 ROOT1=-X/2.+SQRT(QUOT)/2.
ROOT2=-X/2.-SQRT(QUOT)/2.
MU(1)=CMPLX(ROCT1,0.)
MU(2)=CMPLX(ROOT2,0.)
GO TO 24
22 MU(1)=CMPLX(-X/2.,0.)
MU(2)=MU(1)
GO TO 24
21 MU(1)=CMPLX(-X/2.,SQRT(-QUOT)/2.)
MU(2)=CMPLX(-X/2.,-SQRT(-QUOT)/2.)
24 QUOT=U**2-4.*V
IF(QUOT) 25,26,27
27 ROOT3=-U/2.+SQRT(QUOT)/2.
RCOT4=-U/2.-SQRT(QUOT)/2.
MU(3)=CMPLX(ROCT3,0.)
MU(4)=CMPLX(ROCT4,0.)
GO TO 28
26 MU(3)=CMPLX(-U/2.,0.)
MU(4)=MU(3)
GO TO 28

```

```

25 MU(3)=CMPLX(-U/2.,SQRT(-QUOT)/2.)
   MU(4)=CMPLX(-U/2.,-SQRT(-QUOT)/2.)
28 IF(AIMAG(MU(1)).EQ.0.) GO TO 30
   SAVE=MU(1)
   MU(1)=MU(3)
   MU(3)=SAVE
   SAVE=MU(2)
   MU(2)=MU(4)
   MU(4)=SAVE
30 CONTINUE
   DO 40 I=1,4
40 EIGEN(I)=CLOG(MU(I))*CMPLX(2.,0.)
   RETURN
   END

```

```

MJG 790
MJG 800
MJG 810
MJG 820
MJG 830
MJG 840
MJG 850
MJG 860
MJG 870
MJG 880
MJG 890
MJG 900
MJG 910
MJG 920

```

```

SUBROUTINE LISTER(NPRINT)
COMMON /NAMES/NN(20),PN(11),BETAN(11),DSTAR(11),TPN(11)
1   ,PNF(51),BETANF(51),DSTARF(51),PNIF(51),BETANIF(51),DSTARIF(51),
2   ARIF(51),PNC(51),BETANC(51),DSTARC(51),PNC(51),BETANC(51),DSTARILIS
3   C(51),APN(11),ABETAN(11),ADSTAR(11),BPN(11),BBETAN(11),BDSTAR(11),LIS
4   PHIN(11)
COMMON /PARAM/MU,EIGEN,      C,NCASE,MCASE,NZ,      NPLOT, LIS
1NHIST,PNFINAL,BNFINAL,DSFINAL,TIME(51),P(4,4),Q(4) LIS
COMMON /LIMITS/PNU(51),PNL(51),BETANU(51),BETANL(51),DSTARU(51), LIS
1DSTARL(51),PNU(51),PNL(51),BETANU(51),BETANL(51),DSTARU(51), LIS
2DSTARL(51) LIS
COMMON /FINAL/ PSS,BETASS,DSTARSS,DP,DR,OBETA,DPHI,A(4,4),B(4),ODELIS
1LTA LIS
COMMON MU(4),EIGEN(4),C(5,4) LIS
PRINT 2000 LIS
GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170, LIS
1180,190,200),NPRINT LIS
10 CALL DATE(S) LIS
PRINT 1000,S LIS
PRINT 1010,NN,MCASE,NPLOT,NHIST,PN,TPN,BETAN,      TPN,DSTAR, LIS
GO TO 210 LIS
20 PRINT 1020,MU,EIGEN LIS
GO TO 210 LIS
30 PRINT 1030,MU,EIGEN LIS
GO TO 210 LIS
40 PRINT 1040,PHIN,TPN LIS
GO TO 210 LIS
50 PRINT 1050,C LIS
GO TO 210 LIS
60 PRINT 1060,MU,EIGEN,C LIS
GO TO 210 LIS
70 CONTINUE LIS
80 CONTINUE LIS
GO TO 210 LIS
90 PRINT 1090,(TIME(I),PNU(I),PNL(I),BETANU(I),BETANL(I),DSTARU(I), LIS
1DSTARL(I),PNU(I),PNL(I),BETANU(I),BETANL(I),DSTARU(I),DSTARL LIS
2(I),I=1,51) LIS
GO TO 210 LIS

```

```

100 PRINT 1100, (TIME(I), PNF(I), BETANF(I), DSTARF(I), PNIF(I), BETANIF(I)), LIS 400
    10ST, R1F(I), I=1, 51) LIS 410
    GO TO 210 LIS 420
110 PRINT 1110, NZ LIS 430
    GO TO 210 LIS 440
120 PRINT 1120, ((P(I, J), J=1, 4), I=1, 4) LIS 450
    GO TO 210 LIS 460
130 PRINT 1130, Q LIS 470
    GO TO 210 LIS 480
140 PRINT 1140, (TIME(I), PNC(I), BETANC(I), DSTARC(I), I=1, 51) LIS 490
    GO TO 210 LIS 500
150 PRINT 1150, (TIME(I), PNIC(I), BETANIC(I), DSTARIC(I), I=1, 51) LIS 510
    GO TO 210 LIS 520
160 PRINT 1160 LIS 530
    GO TO 210 LIS 540
170 PRINT 1170 LIS 550
    GO TO 210 LIS 560
180 PRINT 1180 LIS 570
    GO TO 210 LIS 580
190 CONTINUE LIS 590
200 CONTINUE LIS 600
210 RETURN LIS 610
    LIS 620
1000 FORMAT(///, 10X, 5HDATE=, A10, //) LIS 630
1010 FORMAT(10X, 20HLIST PARAMETER TABLE, 20I3//5X12H CASE NUMBER, IS, 10HLIS 640
    1 PLOT CODE, IS, 5X, 19H TIME RESPONSE CODE, IS, //, 20X, 10HINPUT DATA, //LIS 650
    2, 5X, 7HPN, , 11F9.3, /, 5X, 7HTIME, , 11F9.2, //, 5X, 7HBETAN, , 11F9.3, LIS 660
    3, /, 5X, 7HTIME, , 11F9.2, //, 5X, 7HDSSTAR, , 11F9.3, /, 5X, 7HTIME, , 11F9.2, LIS 670
    4, //) LIS 680
1020 FORMAT(5X, 4(4X, 4HREAL, 10X, 4HIMAG, 11X), //, 2X, 3HMU, , 4(2E14.6, 5X), //, LIS 690
    12X, 3HEI, , 4(F10.4, 4X, F10.4, 9X)) LIS 700
1030 FORMAT(5X, 19HMU VALUES SPECIFIED, //, 8X, 4( 4HREAL, 10X, 4HIMAG, 10X) LIS 710
    1, //, 1X, 3HMU, , 4(2E14.6 ), //, 1X, 6HEIGEN, , 4(F10.4, 4X, F10.4, 4X)) LIS 720
1040 FORMAT(5X, 26HTHE GENERATED PHI DATA IS, //, 5X, 7MPHN, , 11F9.3, LIS 730
    1, /, 5X, 7HTIME, , 11F9.2) LIS 740
1050 FORMAT(5X, 26HTIME RESPONSE COEFFICIENTS, //, 12X, 5(4X, 4HREAL, 6X, 4HIMLIS 750
    1AG, 5X), //, 5X, 6HPN, , 5(2F10.4, 3X), //, 5X, 6HBETAN, , 5(2F10.4, 3X), //, LIS 760
    25X, 6MPHN, , 5(2F10.4, 3X), //, 5X, 6HDSSTAR, , 5(2F10.4, 3X)) LIS 770
1060 FORMAT(5X, 15HFITTING RESULTS, //, 5X, 5(11X, 4HREAL, 6X, 4HIMAG) , //, LIS 780

```



```

SUBROUTINE GRAPHS
COMMON /NAMES/NN(20),PN(11),BETAN(11),DSTAR(11),TPN(11)
1  ,PNF(51),BETANF(51),DSTARF(51),PNIF(51),BETANIF(51),DSTARGRA
2AR1F(51),PNC(51),BETANC(51),DSTARC(51),PNIC(51),BETANIC(51),DSTAR16RA
3C(51),APN(11),ABETAN(11),ADSTAR(11),BPN(11),BBETAN(11),BDSTAR(11),GRA
4PHIN(11)
COMMON /PARAM/MU,EIGEN, C,NCASE,MCASE, NZ, NPLOT,
1NHIST,PNFINAL,BNFINAL,OSFINAL,TIME(51),P(4,4),Q(4)
COMMON /LIMITS/PNU(51),PNL(51),BETANU(51),BETANL(51),DSTARU(51),
1DSTARL(51),PNUU(51),PNLU(51),BETANUU(51),BETANLU(51),DSTARUU(51),
2DSTARLU(51)
COMMON /FINAL/ PSS,BETASS,DSTARSS,OP,OR,OBETA,OPHI,A(4,4),B(4),DOE3RA
1LTA
COMMON MU(4),EIGEN(4),C(5,4)
IF(NPLOT.EQ.1) GO TO 10
IF(NPLOT.EQ.3) GO TO 10
IF(NPLOT.EQ.6) GO TO 10
IF(NPLOT.EQ.7) GO TO 20
10 IF(NN(16).EQ.1) CALL LISTER(16)
CALL QIKSET(5.0,0.0,0.0,3.0,0.0,0.5)
CALL QIKPLT(TIME,PNU,51,14H$TIME SECONDS$,14H$PN 1/SECONDS$,11H$ROGRA
1LL RATE$)
CALL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,PNL,51,0)
CALL QLINE(TPN,PN,-11,74)
CALL QLINE(TIME,PNF,51,0)
CALL QLINE(TIME,PNC,51,0)
CALL PLOT(-1.5,3.0,-3)
CALL QIKSET(5.0,0.0,0.0,3.0,-2.0,2.0)
CALL QIKPLT(TIME,PNU,51,14H$TIME SECONDS$,8H$CPN/DT$,3H$ S)
CA LL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,PNU,51,0)
CALL QLINE(TIME,PNIF,51,0)
CALL QLINE(TIME,PNIC,51,0)
CALL ENDPLT
20 CONTINUE
IF(NPLOT.EQ.2) GO TO 21
IF(NPLOT.EQ.3) GO TO 21
IF(NPLOT.EQ.5) GO TO 21

```

```

IF(IMPLOT.NE.7) GO TO 30
21 IF(NN(17).EQ.1) CALL LISTER(17)
CALL QIKSET(5.0,0.0,0.0,3.0,-5.0,4.0)
CALL QIKPLT(TIME,BETANU,51,14H$TIME SECONDS$,17H$BETAN 1/SECONDS$,32A
110H$SIDESLIP$)
CALL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,BETANL,51,0)
CALL QLINE(TIME,BETANF,51,0)
CALL QLINE(TPN ,BETAN,-11,74)
CALL QLINE(TIME,BETANC,51,0)
CALL PLOT(-1.5,3.0,-3)
CALL QIKSET(5.0,0.0,0.0,3.0,-6.0,4.0)
CALL QIKPLT(TIME,BETAN1U,51,14H$TIME SECONDS$,10H$DBETA/DT$,3H$ B)
CALL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,BETAN1L,51,0)
CALL QLINE(TIME,BETAN1F,51,0)
CALL QLINE(TIME,BETAN1C,51,0)
CALL ENDPLT
30 CONTINUE
IF(IMPLOT.LT.4) GO TO 40
IF(NN(18).EQ.1) CALL LISTER(18)
CALL QIKSET(5.0,0.0,0.0,4.0,-10.0,5.0)
CALL QIKPLT(TIME,DSTARU,51,14H$TIME SECONDS$,17H$DSTAR 1/SECONDS$,32A
113H$LATL. CRIT.$)
CALL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,DSTARL,51,0)
CALL QLINE(TIME,DSTARF,51,0)
CALL QLINE(TPN ,DSTAR,-11,74)
CALL QLINE(TIME,DSTARC,51,0)
CALL PLOT(-1.5,4.0,-3)
CALL QIKSET(5.0,0.0,0.0,3.0,-12.0,8.0)
CALL QIKPLT(TIME,DSTAR1U,51,14H$TIME SECONDS$,11H$DDSTAR/DT$,3H$ S
1)
CALL PLOT(-6.0,1.0,-3)
CALL QLINE(TIME,DSTAR1L,51,0)
CALL QLINE(TIME,DSTAR1F,51,0)
CALL QLINE(TIME,DSTAR1C,51,0)
CALL ENDPLT

```


GRA 780
GRA 790
GRA 800

40 CONTINUE
RETURN
END

```

SUBROUTINE CASE1(MU, Q, EIGEN, C)
  DIMENSION D(11), U(3, 6), UINV(6, 8), CC(8), PP(8)
  DIMENSION UU(6, 6), UUIINV(6, 6), PPPP(6), CCC(6)
  DIMENSION UUU(4, 4), UUIINV(4, 4), PPPPPP(4), CCCC(4)
  COMPLEX MU(4), DD(10), Q(4, 4), P(4), FC(10, 4), C(5), EIGEN(4)
  COMPLEX QQ(3, 3), PPP(3), RRC(10, 3)
  COMPLEX QQQ(2, 2), PPPPP(2), RRRRC(10, 2)
  COMPLEX A1, A2, A3, A4
  DO 1 I=1, 10
  DO 1 J=1, 4
  1 RC(I, J) = (MU(J) * I - 1.)
  DO 2 I=1, 10
  2 DD(I) = CMPLX((D(I+1) - D(I)), 0.)
  DO 3 I=1, 4
  DO 3 J=1, 4
  Q(I, J) = 0.
  DO 3 K=1, 10
  3 Q(I, J) = Q(I, J) + RC(K, J) * RC(K, I)
  DO 4 I=1, 4
  P(I) = 0.
  DO 4 K=1, 10
  4 P(I) = P(I) + DD(K) * RC(K, I)
  DO 5 I=1, 4
  DO 5 J=1, 4
  U(I, J) = REAL(Q(I, J))
  U(I+4, J+4) = REAL(Q(I, J))
  U(I+4, J) = AIMAG(Q(I, J))
  U(I, J+4) = -AIMAG(Q(I, J))
  5 CALL INVR(U, UINV, 8, 0, 8)
  DO 6 I=1, 4
  PP(I) = REAL(P(I))
  6 PP(I+4) = AIMAG(P(I))
  DO 7 I=1, 8
  CC(I) = 0.
  DO 7 J=1, 8
  7 CC(I) = CC(I) + UINV(I, J) * PP(J)
  DO 8 I=1, 4
  8 C(I) = CMPLX(CC(I), CC(I+4))
  C(5) = -C(1) - C(2) - C(3) - C(4)

```

CAS 10
 CAS 20
 CAS 30
 CAS 40
 CAS 50
 CAS 60
 CAS 70
 CAS 80
 CAS 90
 CAS 100
 CAS 110
 CAS 120
 CAS 130
 CAS 140
 CAS 150
 CAS 160
 CAS 170
 CAS 180
 CAS 190
 CAS 200
 CAS 210
 CAS 220
 CAS 230
 CAS 240
 CAS 250
 CAS 260
 CAS 270
 CAS 280
 CAS 290
 CAS 300
 CAS 310
 CAS 320
 CAS 330
 CAS 340
 CAS 350
 CAS 360
 CAS 370
 CAS 380
 CAS 390

```

      RETURN
      ENTRY CASE2
      A2=-EIGEN(2)/EIGEN(1)
      A3=-EIGEN(3)/EIGEN(1)
      A4=-EIGEN(4)/EIGEN(1)
      DO 10 I=1,10
      RRC(I,1)=1.+A2-MU(2)**I-A2*MU(1)**I
      RRC(I,2)=1.+A3-MU(3)**I-A3*MU(1)**I
      RRC(I,3)=1.+A4-MU(4)**I-A4*MU(1)**I
      DO 20 I=1,10
      DD(I)=CMPLX(D(I+1),0.)
      DO 30 I=1,3
      DO 30 J=1,3
      QQ(I,J)=0.
      DO 30 K=1,10
      QQ(I,J)=QQ(I,J)-RRC(K,J)*RRC(K,I)
      DO 40 I=1,3
      PPP(I)=0.
      DO 40 K=1,10
      PPP(I)=PPP(I)+DD(K)*RRC(K,I)
      DO 50 I=1,3
      DO 50 J=1,3
      UU(I,J)=REAL(QQ(I,J))
      UU(I+3,J+3)=REAL(QQ(I,J))
      UU(I+3,J)=AIMAG(QQ(I,J))
      UU(I,J+3)=-AIMAG(QQ(I,J))
      CALL INVR(UU,UUINV,6,0,6)
      DO 60 I=1,3
      PPPP(I)=REAL(PPP(I))
      PPPP(I+3)=AIMAG(PPP(I))
      DO 70 I=1,6
      CCG(I)=0.
      DO 70 J=1,6
      CCC(I)=CCG(I)+UUINV(I,J)*PPPP(J)
      DO 80 I=1,3
      C(I+1)=CMPLX(CCG(I),CCC(I+3))
      C(1)=A2*C(2)+A3*C(3)+A4*C(4)
      C(5)=-C(1)-C(2)-C(3)-C(4)
      RETURN

```

```

CAS 400
CAS 410
CAS 420
CAS 430
CAS 440
CAS 450
CAS 460
CAS 470
CAS 480
CAS 490
CAS 500
CAS 510
CAS 520
CAS 530
CAS 540
CAS 550
CAS 560
CAS 570
CAS 580
CAS 590
CAS 600
CAS 610
CAS 620
CAS 630
CAS 640
CAS 650
CAS 660
CAS 670
CAS 680
CAS 690
CAS 700
CAS 710
CAS 720
CAS 730
CAS 740
CAS 750
CAS 760
CAS 770
CAS 780

```

```

ENTRY CASE3
A1=EIGEN(3)*(EIGEN(3)-EIGEN(2))/(EIGEN(1)*(EIGEN(2)-EIGEN(1)))
A2=EIGEN(4)*(EIGEN(4)-EIGEN(2))/(EIGEN(1)*(EIGEN(2)-EIGEN(1)))
A3=EIGEN(3)*(EIGEN(3)-EIGEN(1))/(EIGEN(2)*(EIGEN(1)-EIGEN(2)))
A4=EIGEN(4)*(EIGEN(4)-EIGEN(1))/(EIGEN(2)*(EIGEN(1)-EIGEN(2)))
DO 100 I=1,10
  RRRRC(I,1)=1.+A3+A1-MU(3)**I-A3*MU(2)**I-A1*MU(1)**I
  100 RRRRC(I,2)=1.+A4+A2-MU(4)**I-A4*MU(2)**I-A2*MU(1)**I
DO 200 I=1,10
  200 DD(I)=CMPLX(D(I+1),0.)
DO 300 I=1,2
DO 300 J=1,2
  QQQ(I,J)=0.
DO 300 K=1,10
  300 QQQ(I,J)=QQQ(I,J)-RRRC(K,J)*RRRC(K,I)
DO 400 I=1,2
  PPPPP(I)=0.
DO 400 K=1,10
  400 PPPPP(I)=PPPP(I)+DD(K)*RRRC(K,I)
C(3)=(QQQ(2,2)*PPPP(1)-QQQ(1,2)*PPPP(2))/(QQQ(1,1)*QQQ(2,2)-
1 QQQ(1,2)*QQQ(2,1))
C(4)=(QQQ(1,1)*PPPP(2)-QQQ(2,1)*PPPP(1))/(QQQ(1,1)*QQQ(2,2)-
1 QQQ(1,2)*QQQ(2,1))
C(1)=A1*C(3)+A2*C(4)
C(2)=A3*C(3)+A4*C(4)
C(5)=-C(1)-C(2)-C(3)-C(4)
RETURN
END

```

```

CAS 790
CAS 800
CAS 810
CAS 820
CAS 830
CAS 840
CAS 850
CAS 860
CAS 870
CAS 880
CAS 890
CAS 900
CAS 910
CAS 920
CAS 930
CAS 940
CAS 950
CAS 960
CAS 970
CAS 980
CAS 990
CAS 1000
CAS 1010
CAS 1020
CAS 1030
CAS 1040
CAS 1050
CAS 1060

```

```

C      SUBROUTINE INVR(A,B,JJJ,IT,MX)
C      1MX,MV,MU,MS,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6)
C      PROGRAM AUTHORS R.E. FUNDERIC AND R.G. EDWARDS,
C      COMPUTING TECHNOLOGY CENTER, UNION CARBIDE CORP., NUCLEAR DIV.,
C      OAK RIDGE, TENN.
C
C      CTC ORD PROGRAM NO. 9067.1
C      DIMENSION A(MX,MX),B(MX,MX)
C      MAT1=MX
C      MAT2=MX
C      IF (JJJ.NE.1) GO TO 50
C      A(1,1)=1./A(1,1)
C      RETURN
C      50 CONTINUE
C      DO 21 I=1,JJJ
C      DO 20 J=1,JJJ
C      B(I,J)=0.0
C      20 CONTINUE
C      B(I,I)=1.0
C      21 CONTINUE
C      KK=JJJ
C      NV=JJJ
C      D=1.
C      IF (JJJ.LT.0)D=0.
C      KKM=KK-1
C      DO 9 I=1,KKM
C      S=0.0
C      DO 1 J=I,KK
C      R=ABS(A(J,I))
C      IF (R.LT.S) GO TO 1
C      S=R
C      L=J
C      1 CONTINUE
C      IF (L.EQ.I) GO TO 5
C      DO 2 J=I,KK
C      S=A(I,J)
C      A(I,J)=A(L,J)
C      A(L,J)=S
C      2 CONTINUE

```

```

14V 10
14V 20
14V 30
14V 40
14V 50
14V 60
14V 70
14V 80
14V 90
14V 100
14V 110
14V 120
14V 130
14V 140
14V 150
14V 160
14V 170
14V 180
14V 190
14V 200
14V 210
14V 220
14V 230
14V 240
14V 250
14V 260
14V 270
14V 280
14V 290
14V 300
14V 310
14V 320
14V 330
14V 340
14V 350
14V 360
14V 370
14V 380
14V 390

```

INV 400
 INV 410
 INV 420
 INV 430
 INV 440
 INV 450
 INV 460
 INV 470
 INV 480
 INV 490
 INV 500
 INV 510
 INV 520
 INV 530
 INV 540
 INV 550
 INV 560
 INV 570
 INV 580
 INV 590
 INV 600
 INV 610
 INV 620
 INV 630
 INV 640
 INV 650
 INV 660
 INV 670
 INV 680
 INV 690
 INV 700
 INV 710
 INV 720
 INV 730
 INV 740
 INV 750
 INV 760
 INV 770
 INV 780
 INV 790

```

    IF(NV.LE.0)GO TO 4
    DO 3 J=1,NV
      S=B(I,J)
      B(I,J)=B(L,J)
      B(L,J)=S
    3 CONTINUE
    4 D=-D
    5 IF(A(I,I).EQ.0.)GO TO 9
      IPO=I+1
      DO 8 J=IPO,KK
        IF (A(J,I).EQ.0.) GO TO 8
        S=A(J,I)/A(I,I)
        A(J,I)=0.0
        DO 6 K=IPO,KK
          A(J,K)=A(J,K)-A(I,K)*S
        6 CONTINUE
      IF (NV.LE.0) GO TO 8
      DO 7 K=1,NV
        B(J,K)=B(J,K)-B(I,K)*S
      7 CONTINUE
    8 CONTINUE
    9 CONTINUE
    DO 10 I=1,KK
      D=D*A(I,I)
    10 CONTINUE
    IF(NV.LE.0)GO TO 13
    KMO=KK-1
    DO 12 K=1,NV
      B(KK,K)=R(KK,K)/A(KK,KK)
      DO 12 I=1,KMO
        N=KK-I
        DO 11 J=N,KMO
          B(N,K)=B(N,K)-A(N,J+1)*B(J+1,K)
        11 CONTINUE
        B(N,K)=B(N,K)/A(N,N)
      12 CONTINUE
    13 DMATEQ=D
    IF (IT.EQ.0) RETURN
    DO 30 I=1,JJJ
      DO 30 J=1,JJJ

```

IF (ABS(B(I,J)).LT.1.E-5)B(I,J)=0.
30 CONTINUE
RETURN
END

INV 000
INV 010
INV 020
INV 030

```

SUBROUTINE RESPON5
COMMON /NAMES/NN(20),PN(11),BETAN(11),DSTAR(11),TPN(11)
1      ,PNF(51),BETANF(51),DSTARF(51),PN1F(51),BETAN1F(51),DSTAR1F(51)
2      ,PNC(51),BETANC(51),DSTARC(51),PN1C(51),BETAN1C(51),DSTAR1C(51)
3      ,APN(11),ABETAN(11),ADSTAR(11),BPN(11),BBETAN(11),BDSTAR(11),RES
4      ,PHIN(11)
COMMON /PARAM/MU,EIGEN,      C,NCASE,MCASE,NZ,      MNPLOT,
1NHIST,PNFINAL,BNFINAL,DSFINAL,TIME(51),P(4,4),Q(4)
COMMON /LIMITS/PNU(51),PNL(51),BETANU(51),BETANL(51),DSTARU(51),
1DSTARL(51),PNU(51),PN1L(51),BETAN1U(51),BETAN1L(51),DSTAR1U(51),
2DSTAR1L(51)
COMMON /FINAL/ PSS,BETASS,DSTARSS,DP,DR,DBETA,DPHI,A(4,4),B(4),DDERES
1LTA
COMPLEX MU(4),EIGEN(4),C(5,4)
DIMENSION TEMP(4,4),XTEMP(4,4),YTEMP(4,4),R(4,4)
DIMENSION XR1(51),XR2(51),XR3(51),XR4(51)
DO 1 I=1,51
1  TIME(I)=FLOAT(I-1)/10.
  IF(NN(9).EQ.1) CALL LISTER(9)
  DO 10 I=1,51
    PNF(I)=C(1,1)*CEXP(EIGEN(1)*TIME(I))+C(2,1)*CEXP(EIGEN(2)*TIME(I))
    1+C(3,1)*CEXP(EIGEN(3)*TIME(I))+C(4,1)*CEXP(EIGEN(4)*TIME(I))+C(5,1)
    2)
    BETANF(I)=C(1,2)*CEXP(EIGEN(1)*TIME(I))+C(2,2)*CEXP(EIGEN(2)*TIME(I))
    1(I))+C(3,2)*CEXP(EIGEN(3)*TIME(I))+C(4,2)*CEXP(EIGEN(4)*TIME(I))+
    2C(5,2)
    DSTARF(I)=C(1,4)*CEXP(EIGEN(1)*TIME(I))+C(2,4)*CEXP(EIGEN(2)*TIME(I))
    1(I))+C(3,4)*CEXP(EIGEN(3)*TIME(I))+C(4,4)*CEXP(EIGEN(4)*TIME(I))+
    2C(5,4)
    PN1F(I)=EIGEN(1)*C(1,1)*CEXP(EIGEN(1)*TIME(I))+EIGEN(2)*C(2,1)
    1*CEXP(EIGEN(2)*TIME(I))+EIGEN(3)*C(3,1)*CEXP(EIGEN(3)*TIME(I))+
    2EIGEN(4)*C(4,1)*CEXP(EIGEN(4)*TIME(I))
    BETAN1F(I)=EIGEN(1)*C(1,2)*CEXP(EIGEN(1)*TIME(I))+EIGEN(2)*C(2,2)
    1*CEXP(EIGEN(2)*TIME(I))+EIGEN(3)*C(3,2)*CEXP(EIGEN(3)*TIME(I))+
    2EIGEN(4)*C(4,2)*CEXP(EIGEN(4)*TIME(I))
    DSTAR1F(I)=EIGEN(1)*C(1,4)*CEXP(EIGEN(1)*TIME(I))+EIGEN(2)*C(2,4)
    1*CEXP(EIGEN(2)*TIME(I))+EIGEN(3)*C(3,4)*CEXP(EIGEN(3)*TIME(I))+
    2EIGEN(4)*C(4,4)*CEXP(EIGEN(4)*TIME(I))
10 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

C
C
C
IF(NN(10).EQ.1) CALL LISTER(10)
PAYNTERS RECIPE
AA=ABS(A(1))
DO 20 J=2,16
IF(ABS(A(J)).GT.ABS(AA)) AA=ABS(A(J))
20 CONTINUE
AAAT=0.1*AA
FACT=1.
DO 30 K=1,30
FACT=FACT*FLOAT(K)
X=(4.*AAAT)**K*EXP(4.*AAAT)/FACT
IF(X.LT.0.001) GO TO 40
30 NZ=K+1
40 CONTINUE
IF(NN(11).EQ.1) CALL LISTER(11)
TEMP(1)=TEMP(5)=TEMP(11)=TEMP(16)=1.
TEMP(2)=TEMP(3)=TEMP(4)=TEMP(5)=TEMP(7)=TEMP(8)=TEMP(9)=TEMP(10)=0.
TEMP(12)=TEMP(13)=TEMP(14)=TEMP(15)=0.
P(1,1)=P(2,2)=P(3,3)=P(4,4)=1.
P(1,2)=P(1,3)=P(1,4)=P(2,1)=P(2,3)=P(2,4)=P(3,1)=P(3,2)=P(3,4)=0.
P(4,1)=P(4,2)=P(4,3)=0.
DO 50 I=1,NZ
DO 51 JJ=1,16
51 XTEMP(JJ)=0.0
DO 55 J=1,4
DO 55 K=1,4
DO 55 L=1,4
55 XTEMP(J,K)=XTEMP(J,K)+TEMP(J,L)*A(L,K)*0.1/FLOAT(I)
DO 52 JJ=1,16
52 TEMP(JJ)=XTEMP(JJ)
DO 60 II=1,16
60 P(II)=P(II)+TEMP(II)
50 CONTINUE
IF(NN(12).EQ.1) CALL LISTER(12)
TEMP(1)=TEMP(6)=TEMP(11)=TEMP(16)=1.
TEMP(2)=TEMP(3)=TEMP(4)=TEMP(5)=TEMP(7)=TEMP(8)=TEMP(9)=TEMP(10)=0.
TEMP(12)=TEMP(13)=TEMP(14)=TEMP(15)=0.

```

```

RES 400
RES 410
RES 420
RES 430
RES 440
RES 450
RES 460
RES 470
RES 480
RES 490
RES 500
RES 510
RES 520
RES 530
RES 540
RES 550
RES 560
RES 570
RES 580
RES 590
RES 600
RES 610
RES 620
RES 630
RES 640
RES 650
RES 660
RES 670
RES 680
RES 690
RES 700
RES 710
RES 720
RES 730
RES 740
RES 750
RES 760
RES 770
RES 780

```

```

R(1,1)=R(2,2)=R(3,3)=R(4,4)=1.
R(1,2)=R(1,3)=R(1,4)=R(2,1)=R(2,3)=R(2,4)=R(3,1)=R(3,2)=R(3,4)=0.
R(4,1)=R(4,2)=R(4,3)=0.
DO 61 I=1,NZ
DO 62 JJ=1,16
62 XTEMP(JJ)=0.0
DO 65 J=1,4
DO 55 K=1,4
DO 65 L=1,4
65 XTEMP(J,K)=XTEMP(J,K)+TEMP(J,L)*A(L,K)*0.1/FLOAT(I+1)
DO 67 JJ=1,16
67 TEMP(JJ)=XTEMP(JJ)
DO 69 II=1,16
69 R(II)=R(II)+TEMP(II)
61 CONTINUE
Q(1)=R(1,1)*B(1,1)+R(1,2)*B(2,1)+R(1,3)*B(3,1)+R(1,4)*B(4,1)
Q(2)=R(2,1)*B(1,1)+R(2,2)*B(2,1)+R(2,3)*B(3,1)+R(2,4)*B(4,1)
Q(3)=R(3,1)*B(1,1)+R(3,2)*B(2,1)+R(3,3)*B(3,1)+R(3,4)*B(4,1)
Q(4)=R(4,1)*B(1,1)+R(4,2)*B(2,1)+R(4,3)*B(3,1)+R(4,4)*B(4,1)
Q(1)=Q(1)*0.1
Q(2)=Q(2)*0.1
Q(3)=Q(3)*0.1
Q(4)=Q(4)*0.1
IF(MN(13).EQ.1) CALL LISTER(13)
XR1(1)=XR3(1)=XR4(1)=0.
XR2(1)=-DDelta/DR
DO 70 I=2,51
XR1(I)=P(1,1)*XR1(I-1)+P(1,2)*XR2(I-1)+P(1,3)*XR3(I-1)+P(1,4)*XR4(I-1)+Q(1)
XR2(I)=P(2,1)*XR1(I-1)+P(2,2)*XR2(I-1)+P(2,3)*XR3(I-1)+P(2,4)*XR4(I-1)+Q(2)
XR3(I)=P(3,1)*XR1(I-1)+P(3,2)*XR2(I-1)+P(3,3)*XR3(I-1)+P(3,4)*XR4(I-1)+Q(3)
XR4(I)=P(4,1)*XR1(I-1)+P(4,2)*XR2(I-1)+P(4,3)*XR3(I-1)+P(4,4)*XR4(I-1)+Q(4)
DO 71 I=1,51
PNC(I)=PSS*XR1(I)
SETANC(I)=BETASS*XR3(I)
71 DISTARC(I)=DP*XR1(I)+DR*XR2(I)+DBETA*XR3(I)+DPHI*XR4(I)+DDelta

```

```

RES 790
RES 800
RES 810
RES 820
RES 830
RES 840
RES 850
RES 860
RES 870
RES 880
RES 890
RES 900
RES 910
RES 920
RES 930
RES 940
RES 950
RES 960
RES 970
RES 980
RES 990
RES 1000
RES 1010
RES 1020
RES 1030
RES 1040
RES 1050
RES 1060
RES 1070
RES 1080
RES 1090
RES 1100
RES 1110
RES 1120
RES 1130
RES 1140
RES 1150
RES 1160
RES 1170

```

```

IF(NN(14).EQ.1) CALL LISTER(14)
DO 80 I=1,51
  XR1T=XR1(I)*A(1,1)+XR2(I)*A(1,2)+XR3(I)*A(1,3)+XR4(I)*A(1,4)+B(1)
  XR2T=XR1(I)*A(2,1)+XR2(I)*A(2,2)+XR3(I)*A(2,3)+XR4(I)*A(2,4)+B(2)
  XR3T=XR1(I)*A(3,1)+XR2(I)*A(3,2)+XR3(I)*A(3,3)+XR4(I)*A(3,4)+B(3)
  XR4T=XR1(I)*A(4,1)+XR2(I)*A(4,2)+XR3(I)*A(4,3)+XR4(I)*A(4,4)+B(4)
  PNIC(I)=PSS*XR1T
  BETANIC(I)=BETASS*XR3T
  80 DSTARIC(I)=DP*XR1T+DR*XR2T+DBETA*XR3T+DPHI*XR4T
  IF(NN(15).EQ.1) CALL LISTER(15)
  RETURN
END
RES 1180
RES 1190
RES 1200
RES 1210
RES 1220
RES 1230
RES 1240
RES 1250
RES 1260
RES 1270
RES 1280
RES 1290

```

```

SUBROUTINE MODEL(C,EIGEN,PSS,BETASS,DSS
1DDelta)
COMPLEX C(5,4),EIGEN(4)
DIMENSION A(4,4),B(4),X(16),XINC(16),P(16,16),PINV(16,16),F(16),
11T(4,4),AT(4,4),M(4)
REAL L,L1,L2,K,I31,I32,I33,I34
DATA X/1.,4*0.,1.,4*0.,1.,4*0.,1./
READ 100,V,L,C3,QCQ,PSS,BETASS,DSS
PRINT 200,V,L,C3,QCQ,PSS,BETASS,DSS
READ 110,ITMAX,EPS1
PRINT 300,ITMAX,EPS1,X
K=C3*QCQ
L1=REAL(EIGEN(1))
L2=REAL(EIGEN(2))
A3=REAL(EIGEN(3))
B3=AIMAG(EIGEN(3))
R11=REAL(C(1,1))/PSS
R12=REAL(C(1,2))/BETASS
R13=REAL(C(1,3))/PSS
R14=REAL(C(1,4))/DSS
R21=REAL(C(2,1))/PSS
R22=REAL(C(2,2))/BETASS
R23=REAL(C(2,3))/PSS
R24=REAL(C(2,4))/DSS
R31=REAL(C(3,1))/PSS
R32=REAL(C(3,2))/BETASS
R33=REAL(C(3,3))/PSS
R34=REAL(C(3,4))/DSS
I31=AIMAG(C(3,1))/PSS
I32=AIMAG(C(3,2))/BETASS
I33=AIMAG(C(3,3))/PSS
I34=AIMAG(C(3,4))/DSS
C1=V*R11
C2=C1*L/V
C3=V*R12
C4=C3*L/V
C5=V*R13
C6=C5*L/V
C7=(R14-K*R12)

```

M00	400
M00	410
M00	420
M00	430
M00	440
M00	450
M00	460
M00	470
M00	480
M00	490
M00	500
M00	510
M00	520
M00	530
M00	540
M00	550
M00	560
M00	570
M00	580
M00	590
M00	600
M00	610
M00	620
M00	630
M00	640
M00	650
M00	660
M00	670
M00	680
M00	690
M00	700
M00	710
M00	720
M00	730
M00	740
M00	750
M00	760
M00	770
M00	780

C8=-V*R11*L1
 C9=C8*L/V
 D1=-C8
 C10=V*R21
 C11=C10*L/V
 C12=V*R22
 C13=C12*L/V
 C14=V*R23
 C15=C14*L/V
 C16=(R24-K*R22)
 C17=-V*R21*L2
 C18=C17*L/V
 D2=-C17
 C19=V*R31
 C20=C19*L/V
 C21=V*R32
 C22=C21*L/V
 C23=V*R33
 C24=C23*L/V
 C25=R34-K*R32
 C26=V*(I31*B3-R31*A3)
 C27=C26*L/V
 D3=-C26
 C28=V*I31
 C29=C28*L/V
 C30=V*I32
 C31=C30*L/V
 C32=V*I33
 C33=C32*L/V
 C34=I34-K*I32
 C35=-V*(R31*B3+I31*A3)
 C36=C35*L/V
 D4=-C35
 C37=V*R11+L*R11*L1
 C38=V*R12+L*R12*L1
 C39=V*R13+L*R13*L1
 C40=V*R11*L1
 C41=V*S12*L1
 C42=V*R13*L1

D5=R14*L1-K*R12*L1	MOD	790
C43=V*R21+L*R21*L2	MOD	800
C44=V*R22+L*R22*L2	MOD	810
C45=V*R23+L*R23*L2	MOD	820
C46=V*R21*L2	MOD	830
C47=V*R22*L2	MOD	840
C48=V*R23*L2	MOD	850
D6=R24*L2-K*R22*L2	MOD	860
C49=V*R31+L*R31*A3-L*I31*B3	MOD	870
C50=V*R32+L*R32*A3-L*I32*B3	MOD	880
C51=V*R33+L*R33*A3-L*I33*B3	MOD	890
C52=V*R31*A3-V*I31*B3	MOD	900
C53=V*(R32*A3-I32*B3)	MOD	910
C54=V*(R33*A3-I33*B3)	MOD	920
D7= R34*A3-I34*B3-K*R32*A3+K*I32*B3	MOD	930
C55=V*I31+L*I31*A3+L*R31*B3	MOD	940
C56=V*I32+L*R32*B3+L*I32*A3	MOD	950
C57=V*I33+L*R33*B3+L*I33*A3	MOD	960
C58=V*R31*B3+V*I31*A3	MOD	970
C59=V*(R32*B3+I32*A3)	MOD	980
C60=V*(R33*B3+I33*A3)	MOD	990
D8=R34*B3+I34*A3-K*R32*B3-K*I32*A3	MOD	1000
C61=R14-K*R12-V*R12*L1	MOD	1010
C62=-L*R12*L1	MOD	1020
D9=-C62*V/L	MOD	1030
C63=R24-K*R22-V*L2*R22	MOD	1040
C64=-L*L2*R22	MOD	1050
D10=-C64*V/L	MOD	1060
C65=R34-K*R32-V*R32*A3+V*I32*B3	MOD	1070
C66=L*(-R32*A3+I32*B3)	MOD	1080
D11=-C66*V/L	MOD	1090
C67=I34-K*I32-V*I32*A3-V*R32*B3	MOD	1100
C68=-L*(R32*B3+I32*A3)	MOD	1110
D12=-C68*V/L	MOD	1120
C69=-V*R13*L1	MOD	1130
C70=C69*L/V	MOD	1140
D13=-C69	MOD	1150
C71=-V*R23*L2	MOD	1160
C72=C71*L/V	MOD	1170
D14=-C71	MOD	1180

```

C73=-V*(R33*A3-I33*B3)
C74=C73*L/V
D15=-C73
C75=-V*(R33*B3+I33*A3)
C76=C75*L/V
D16=-C75
DO 4 ITER=1,ITMAX
AA=X(1)*X(10)-X(2)*X(9)
AB=X(1)*X(6)-X(2)*X(5)
AC=X(3)*X(10)-X(2)*X(11)
AD=X(3)*X(6)-X(2)*X(7)
AE=X(4)*X(10)-X(2)*X(12)
AF=X(4)*X(6)-X(2)*X(8)
F(1)=C1*AA+C2*AB+C3*AC+C4*AD+C5*AE+C6*AF+C1*X(1)+C7*X(2)+C3*X(3)+
1C5*X(4)+C8*X(10)+C9*X(6)-D1
F(2)=C10*AA+C11*AB+C12*AC+C13*AD+C14*AE+C15*AF+C10*X(1)+C16*X(2)+
1C12*X(3)+C14*X(4)+C17*X(10)+C18*X(6)-D2
F(3)=C19*AA+C20*AB+C21*AC+C22*AD+C23*AE+C24*AF+C19*X(1)+C25*X(2)+
1C21*X(3)+C23*X(4)+C26*X(10)+C27*X(6)-D3
F(4)=C28*AA+C29*AB+C30*AC+C31*AD+C32*AE+C33*AF+C28*X(1)+C34*X(2)+
1C30*X(3)+C32*X(4)+C35*X(10)+C36*X(6)-D4
AA=X(5)*X(10)-X(6)*X(9)
AB=X(7)*X(10)-X(6)*X(11)
AC=X(8)*X(10)-X(6)*X(12)
F(5)=C1*AA+C3*AB+C5*AC+C37*X(5)+C7*X(6)+C38*X(7)+C39*X(8)+C40*X(9)+
1+C41*X(11)+C42*X(12)-D5
F(6)=C10*AA+C12*AB+C14*AC+C43*X(5)+C16*X(6)+C44*X(7)+C45*X(8)+C46*X(9)+
1X(9)+C47*X(11)+C48*X(12)-D6
F(7)=C19*AA+C21*AB+C23*AC+C49*X(5)+C25*X(6)+C50*X(7)+C51*X(8)+C52*X(9)+
1X(9)+C53*X(11)+C54*X(12)-D7
F(8)=C28*AA+C30*AB+C32*AC+C55*X(5)+C34*X(6)+C56*X(7)+C57*X(8)+C58*X(9)+
1X(9)+C59*X(11)+C60*X(12)-D8
AA=X(9)*X(5)-X(5)*X(10)
AB=X(11)*X(6)-X(7)*X(10)
AC=X(12)*X(6)-X(8)*X(10)
F(9)=C2*AA+C4*AB+C6*AC+C1*X(9)+C61*X(10)+C3*X(11)+C5*X(12)+C62*
1X(6)-D9
F(10)=C11*AA+C13*AB+C15*AC+C10*X(9)+C63*X(10)+C12*X(11)+C14*X(12)+
1C64*X(6)-D10

```

```

F(11)=C20*AA+C22*AB+C24*AC+C19*X(9)+C65*X(10)+C21*X(11)+C23*X(12)+MOD 1580
1C6E*X(6)-D11 MOD 1590
F(12)=C29*AA+C31*AB+C33*AC+C28*X(9)+C67*X(10)+C30*X(11)+C32*X(12)+MOD 1600
1C68*X(6)-D12 MOD 1610
AA=X(13)*X(10)-X(14)*X(9) MOD 1620
AB=X(13)*X(6)-X(14)*X(5) MOD 1630
AC=X(15)*X(10)-X(14)*X(11) MOD 1640
AD=X(15)*X(6)-X(14)*X(7) MOD 1650
AE=X(15)*X(10)-X(14)*X(12) MOD 1660
AF=X(16)*X(6)-X(14)*X(8) MOD 1670
F(13)=C1*AA+C2*AB+C3*AC+C4*AD+C5*AE+C6*AF+C1*X(13)+C7*X(14)+C3*
1X(15)+C5*X(16)+C6*X(10)+C70*X(6)-D13 MOD 1680
F(14)=C10*AA+C11*AB+C12*AC+C13*AD+C14*AE+C15*AF+C10*X(13)+C16*
1X(14)+C12*X(15)+C14*X(16)+C71*X(10)+C72*X(6)-D14 MOD 1690
F(15)=C19*AA+C20*AB+C21*AC+C22*AD+C23*AE+C24*AF+C19*X(13)+C25*
1X(14)+C21*X(15)+C23*X(16)+C73*X(10)+C74*X(6)-D15 MOD 1700
F(16)=C28*AA+C29*AB+C30*AC+C31*AD+C32*AE+C33*AF+C28*X(13)+C34*
1X(14)+C30*X(15)+C32*X(16)+C75*X(10)+C76*X(6)-D16 MOD 1710
DO 5 I=1,256 MOD 1720
5 P(I)=0. MOD 1730
P(1,1)=C1*X(10)+C2*X(6)+C1 MOD 1740
P(1,2)=-C1*X(9)-C2*X(5)-C3*X(11)-C4*X(7)-C5*X(12)-C6*X(8)+C7 MOD 1750
P(1,3)=C3*X(10)+C4*X(6)+C3 MOD 1760
P(1,4)=-C5*X(10)+C6*X(6)+C5 MOD 1770
P(1,5)=-C2*X(2) MOD 1780
P(1,6)=C2*X(1)+C4*X(3)+C6*X(4)+C9 MOD 1790
P(1,7)=-C4*X(2) MOD 1800
P(1,8)=-C6*X(2) MOD 1810
P(1,9)=-C1*X(2) MOD 1820
P(1,10)=C1*X(1)+C3*X(3)+C5*X(4)+C8 MOD 1830
P(1,11)=-C3*X(2) MOD 1840
P(1,12)=-C5*X(2) MOD 1850
P(2,1)=C10*X(10)+C11*X(6)+C10 MOD 1860
P(2,2)=-C10*X(9)-C11*X(5)-C12*X(11)-C13*X(7)-C14*X(12)-C15*X(8)+
1C16 MOD 1870
P(2,3)=C12*X(10)+C13*X(6)+C12 MOD 1880
P(2,4)=C14*X(10)+C15*X(6)+C14 MOD 1890
P(2,5)=-C11*X(2) MOD 1900
P(2,6)=C11*X(1)+C13*X(3)+C15*X(4)+C18 MOD 1910
P(2,7)=-C13*X(2) MOD 1920
MOD 1930
MOD 1940
MOD 1950
MOD 1960
MOD 1970

```


$P(2,8) = -C15 * X(2)$
 $P(2,9) = -C10 * X(2)$
 $P(2,10) = C10 * X(1) + C12 * X(3) + C14 * X(4) + C17$
 $P(2,11) = -C12 * X(2)$
 $P(2,12) = -C14 * X(2)$
 $P(3,1) = C19 * X(10) + C20 * X(6) + C19$
 $P(3,2) = -C19 * X(9) - C20 * X(5) - C21 * X(11) - C22 * X(7) - C23 * X(12) - C24 * X(8) +$
 $1C25$
 $P(3,3) = C21 * X(10) + C22 * X(6) + C21$
 $P(3,4) = C23 * X(10) + C24 * X(6) + C23$
 $P(3,5) = -C20 * X(2)$
 $P(3,6) = C20 * X(1) + C22 * X(3) + C24 * X(4) + C27$
 $P(3,7) = -C22 * X(2)$
 $P(3,8) = -C24 * X(2)$
 $P(3,9) = -C19 * X(2)$
 $P(3,10) = C19 * X(1) + C21 * X(3) + C23 * X(4) + C26$
 $P(3,11) = -C21 * X(2)$
 $P(3,12) = -C23 * X(2)$
 $P(4,1) = C28 * X(10) + C29 * X(6) + C28$
 $P(4,2) = -C28 * X(9) - C29 * X(5) - C30 * X(11) - C31 * X(7) - C32 * X(12) - C33 * X(8) +$
 $1C34$
 $P(4,3) = C30 * X(10) + C31 * X(6) + C30$
 $P(4,4) = C32 * X(10) + C33 * X(6) + C32$
 $P(4,5) = -C29 * X(2)$
 $P(4,6) = C29 * X(1) + C31 * X(3) + C33 * X(4) + C36$
 $P(4,7) = -C31 * X(2)$
 $P(4,8) = -C33 * X(2)$
 $P(4,9) = -C28 * X(2)$
 $P(4,10) = C28 * X(1) + C30 * X(3) + C32 * X(4) + C35$
 $P(4,11) = -C30 * X(2)$
 $P(4,12) = -C32 * X(2)$
 $P(5,5) = C1 * X(10) + C37$
 $P(5,6) = -C1 * X(9) - C3 * X(11) - C5 * X(12) + C7$
 $P(5,7) = C3 * X(10) + C38$
 $P(5,8) = C5 * X(10) + C39$
 $P(5,9) = -C1 * X(6) + C40$
 $P(5,10) = C1 * X(5) + C3 * X(7) + C5 * X(8)$
 $P(5,11) = -C3 * X(6) + C41$
 $P(5,12) = -C5 * X(6) + C42$
 $P(6,5) = C10 * X(10) + C43$

MOD 1980
MOD 1990
MOD 2000
MOD 2010
MOD 2020
MOD 2030
MOD 2040
MOD 2050
MOD 2060
MOD 2070
MOD 2080
MOD 2090
MOD 2100
MOD 2110
MOD 2120
MOD 2130
MOD 2140
MOD 2150
MOD 2160
MOD 2170
MOD 2180
MOD 2190
MOD 2200
MOD 2210
MOD 2220
MOD 2230
MOD 2240
MOD 2250
MOD 2260
MOD 2270
MOD 2280
MOD 2290
MOD 2300
MOD 2310
MOD 2320
MOD 2330
MOD 2340
MOD 2350
MOD 2360
MOD 2370

P(6,6)=-C10*X(9)-C12*X(11)-C14*X(12)+C16	MOD 2380
P(6,7)=C12*X(10)+C44	MOD 2390
P(6,8)=C14*X(10)+C45	MOD 2400
P(6,9)=-C10*X(6)+C46	MOD 2410
P(6,10)=C10*X(5)+C12*X(7)+C14*X(8)	MOD 2420
P(6,11)=-C12*X(6)+C47	MOD 2430
P(6,12)=-C14*X(6)+C48	MOD 2440
P(7,5)=C19*X(10)+C49	MOD 2450
P(7,6)=-C19*X(9)-C21*X(11)-C23*X(12)+C25	MOD 2460
P(7,7)=C21*X(10)+C50	MOD 2470
P(7,8)=C23*X(10)+C51	MOD 2480
P(7,9)=-C19*X(6)+C52	MOD 2490
P(7,10)=C19*X(5)+C21*X(7)+C23*X(8)	MOD 2500
P(7,11)=-C21*X(6)+C53	MOD 2510
P(7,12)=-C23*X(6)+C54	MOD 2520
P(8,5)=C28*X(10)+C55	MOD 2530
P(8,6)=-C28*X(9)-C30*X(11)-C32*X(12)+C34	MOD 2540
P(8,7)=C30*X(10)+C56	MOD 2550
P(8,8)=C32*X(10)+C57	MOD 2560
P(8,9)=-C28*X(6)+C58	MOD 2570
P(8,10)=C28*X(5)+C30*X(7)+C32*X(8)	MOD 2580
P(8,11)=-C30*X(6)+C59	MOD 2590
P(8,12)=-C32*X(6)+C60	MOD 2600
P(9,6)=C2*X(9)+C4*X(11)+C6*X(12)+C62	MOD 2610
P(9,5)=-C2*X(10)	MOD 2620
P(9,7)=-C4*X(10)	MOD 2630
P(9,8)=-C6*X(10)	MOD 2640
P(9,9)=C2*X(6)+C1	MOD 2650
P(9,10)=-C2*X(5)-C4*X(7)-C6*X(8)+C61	MOD 2660
P(9,11)=C4*X(6)+C3	MOD 2670
P(9,12)=C6*X(6)+C5	MOD 2680
P(10,5)=-C11*X(10)	MOD 2690
P(10,6)=C11*X(9)+C13*X(11)+C15*X(12)+C64	MOD 2700
P(10,8)=-C15*X(10)	MOD 2710
P(10,7)=-C13*X(10)	MOD 2720
P(10,9)=C11*X(6)+C10	MOD 2730
P(10,10)=-C11*X(5)-C13*X(7)-C15*X(8)+C63	MOD 2740
P(10,11)=C13*X(6)+C12	MOD 2750
P(10,12)=C15*X(6)+C14	MOD 2760

ORIGINAL PAGE IS
OF POOR QUALITY

```

P(11,5)=-C20*X(10)
P(11,6)=C20*X(9)+C22*X(11)+C24*X(12)+C66
P(11,7)=-C22*X(10)
P(11,8)=-C24*X(10)
P(11,9)=C20*X(6)+C19
P(11,10)=-C20*X(5)-C22*X(7)-C24*X(8)+C65
P(11,11)=C22*X(6)+C21
P(11,12)=C24*X(6)+C23
P(12,5)=-C29*X(10)
P(12,6)=C29*X(9)+C31*X(11)+C33*X(12)+C68
P(12,7)=-C31*X(10)
P(12,8)=-C33*X(10)
P(12,9)=C29*X(6)+C28
P(12,10)=-C29*X(5)-C31*X(7)-C33*X(8)+C67
P(12,11)=C31*X(6)+C30
P(12,12)=C33*X(6)+C32
P(13,5)=-C2*X(14)
P(13,6)=C2*X(13)+C4*X(15)+C6*X(16)+C70
P(13,7)=-C4*X(14)
P(13,8)=-C6*X(14)
P(13,9)=-C1*X(14)
P(13,10)=C1*X(13)+C3*X(15)+C5*X(16)+C69
P(13,11)=-C3*X(14)
P(13,12)=-C5*X(14)
P(13,13)=C1*X(10)+C2*X(6)+C1
P(13,14)=-C1*X(9)-C2*X(5)-C3*X(11)-C4*X(7)-C5*X(12)-C6*X(8)+C7
P(13,15)=C3*X(10)+C4*X(6)+C3
P(13,16)=C5*X(10)+C6*X(6)+C5
P(14,5)=-C11*X(14)
P(14,6)=C11*X(13)+C13*X(15)+C15*X(16)+C72
P(14,7)=-C13*X(14)
P(14,9)=-C10*X(14)
P(14,8)=-C15*X(14)
P(14,10)=C10*X(13)+C12*X(15)+C14*X(16)+C71
P(14,11)=-C12*X(14)
P(14,12)=-C14*X(14)
P(14,13)=C10*X(10)+C11*X(6)+C10
P(14,14)=-C10*X(9)-C11*X(5)-C12*X(11)-C13*X(7)-C14*X(12)-C15*X(8)+
1C15

```

```

MOD 2770
MOD 2780
MOD 2790
MOD 2800
MOD 2810
MOD 2820
MOD 2830
MOD 2840
MOD 2850
MOD 2860
MOD 2870
MOD 2880
MOD 2890
MOD 2900
MOD 2910
MOD 2920
MOD 2930
MOD 2940
MOD 2950
MOD 2960
MOD 2970
MOD 2980
MOD 2990
MOD 3000
MOD 3010
MOD 3020
MOD 3030
MOD 3040
MOD 3050
MOD 3060
MOD 3070
MOD 3080
MOD 3090
MOD 3100
MOD 3110
MOD 3120
MOD 3130
MOD 3140
MOD 3150

```

```

P(14,15)=C12*X(10)+C13*X(6)+C12
P(14,16)=C14*X(10)+C15*X(6)+C14
P(15,5)=-C20*X(14)
P(15,6)=C20*X(13)+C22*X(15)+C24*X(16)+C74
P(15,7)=-C22*X(14)
P(15,8)=-C24*X(14)
P(15,9)=-C19*X(14)
P(15,10)=C19*X(13)+C21*X(15)+C23*X(16)+C73
P(15,11)=-C21*X(14)
P(15,12)=-C23*X(14)
P(15,13)=C19*X(10)+C20*X(6)+C19
P(15,14)=-C19*X(9)-C20*X(5)-C21*X(11)-C22*X(7)-C23*X(12)-C24*X(8)+C75
1C25
P(15,15)=C21*X(10)+C22*X(6)+C21
P(15,16)=C23*X(10)+C24*X(6)+C23
P(16,5)=-C29*X(14)
P(16,6)=C29*X(13)+C31*X(15)+C33*X(16)+C76
P(16,7)=-C31*X(14)
P(16,8)=-C33*X(14)
P(16,9)=-C28*X(14)
P(16,10)=C28*X(13)+C30*X(15)+C32*X(16)+C75
P(16,11)=-C30*X(14)
P(16,12)=-C32*X(14)
P(16,13)=C28*X(10)+C29*X(6)+C28
P(16,14)=-C28*X(9)-C29*X(5)-C30*X(11)-C31*X(7)-C32*X(12)-C33*X(8)+C76
1C34
P(16,15)=C30*X(10)+C31*X(6)+C30
P(16,16)=C32*X(10)+C33*X(6)+C32
2 ITCON=1
CALL INVR(P,PINV,16,0,16)
DO 10 I=1,16
XINC(I)=0.
DO 10 J=1,16
10 XINC(I)=XINC(I)-PINV(I,J)*F(J)
DO 3 I=1,16
IF(ABS(XINC(I)).GT.EPS1) ITCON=0
3 X(I)=X(I)+XINC(I)
11 CONTINUE
IF(ITCON.NE.0) GO TO 1

```

```

MOD 3160
MOD 3170
MOD 3180
MOD 3190
MOD 3200
MOD 3210
MOD 3220
MOD 3230
MOD 3240
MOD 3250
MOD 3260
MOD 3270
MOD 3280
MOD 3290
MOD 3300
MOD 3310
MOD 3320
MOD 3330
MOD 3340
MOD 3350
MOD 3360
MOD 3370
MOD 3380
MOD 3390
MOD 3400
MOD 3410
MOD 3420
MOD 3430
MOD 3440
MOD 3450
MOD 3460
MOD 3470
MOD 3480
MOD 3490
MOD 3500
MOD 3510
MOD 3520
MOD 3530
MOD 3540

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

4  CONTINUE
   PRINT 500,X,XINC
   GO TO 7
1  PRINT 400,ITER,X
7  CONTINUE
   DO 6 I=1,4
   DO 6 J=1,4
6  A(I,J)=X(4*I-4+J)
   DP=(V*A(3,1)+L*A(2,1))
   DR=(V*A(3,2)+L*A(2,2)+V)
   DBETA=(V*A(3,3)+L*A(2,3)+K)
   DPHI=(V*A(3,4)+L*A(2,4))
   DO 8 I=1,4
   W(I)=(A(I,1)-A(I,2)*DP/DR)*REAL(C(5,1))/PSS+
1  (A(I,3)-A(I,2)*DBETA/DR)*REAL(C(5,2))/DBETASS+
2  (A(I,4)-A(I,2)*DPHI/DR)*REAL(C(5,3))/PSS+
3  A(I,2)*REAL(C(5,4))/DR/DSS
   DO 8 W(I)=-W(I)
   Z1=1.-L*A(1,2)/DR
   Z2=-V*A(1,2)/DR
   Z3=-L*A(3,2)/DR
   Z4=1.-V*A(3,2)/DR
   B(1)=(Z4*W(1)-Z2*W(3))/(Z1*Z4-Z2*Z3)
   B(3)=(-Z3*W(1)+Z1*W(3))/(Z1*Z4-Z2*Z3)
   DDELTA=(V*B(3)+L*B(1))
   B(2)=W(2)+DDELTA*A(2,2)/DR
   B(4)=W(4)+DDELTA*A(4,2)/DR
   DP=DP+CSS
   DR=DR+DSS
   DBETA=DBETA+DSS
   DPHI=DPHI+DSS
   PRINT 900,PSS,DBETASS,PSS,DP,DR,DBETA,DPHI
   DDELTA=DDELTA+DSS
   PRINT 800,DDELTA
   PRINT 600,((A(I,J),J=1,4),B(I),I=1,4)
   RETURN
100 FORMAT(9F10.0)
110 FORMAT(I4,F20.0)

```

M00 3550
M00 3560
M00 3570
M00 3580
M00 3590
M00 3600
M00 3610
M00 3620
M00 3630
M00 3640
M00 3650
M00 3660
M00 3670
M00 3680
M00 3690
M00 3700
M00 3710
M00 3720
M00 3730
M00 3740
M00 3750
M00 3760
M00 3770
M00 3780
M00 3790
M00 3800
M00 3810
M00 3820
M00 3830
M00 3840
M00 3850
M00 3860
M00 3870
M00 3880
M00 3890
M00 3900
M00 3910
M00 3920

```

200 FORMAT(//,10X,44HAIRPLANE MODEL WITH SPECIFIED TIME HISTORIES,//MOD 3930
1,10X,29MFLIGHT AND VEHICLE PARAMETERS,/,10X,8HAIRSPED,F10.1,7M FMOD 3940
2T/SEC,/,10X,41MCG TO PILOT STATION LONGITUDINAL DISTANCE,F10.2,3HMOD 3958
3 FT,/,10X,39HDIMENSIONAL CONSTANT FOR DSTAR EQUATION,F10.4,30M CUMOD 3960
48IC-FEET/LB-SECONDS-SQUARED,/,10X,16HDYNAMIC PRESSURE,F10.1,14H LMOD 3970
58/FT-SQUARED,/,10X,29MROLLRATE NORMALIZATION FACTOR,F10.3,/,10X,MOD 3980
629MSIDESLIP NORMALIZATION FACTOR,F10.3,/,10X,26MDSTAR NORMALIZATIONMOD 3990
7ON FACTOR,F10.3,/) MOD 4000
300 FORMAT(10X,43HMAXIMUM NUMBER OF NEWTON-RAPHSON ITERATIONS,10,/,10X,MOD 4010
1,17HCONVERGENCE LIMIT,F20.5,MOD 4020
2,/,4H X,16F8.4,/) MOD 4030
400 FCRMAT(//,10X,25HITERATIONS TO CONVERGENCE,10,/,10X,12HN-R SOLUTIONMOD 4040
1ION,/,4X,16F8.4,/) MOD 4050
500 FORMAT(10X,29HN-R SOLUTION DID NOT CONVERGE,/,10X,19HLAST TRIAL SOMOD 4060
1LUTION,/,4X,16F8.4,/,10X,24HLAST SOLUTION ADJUSTMENT,/,4X,16F8.4) MOD 4070
600 FORMAT(10X,15HTHE A MATRIX IS,59X,15HTHE B VECTOR IS,/,4(10X,4E16,MOD 4080
16,10X,E16.6,/) MOD 4090
700 FORMAT(//,10X,20HMODIFIED EIGENMATRIX,/,4(10X,4E20.6,/) MOD 4100
800 FORMAT( 10X,14H(DSTAR)DELTAA=F20.5,/) MOD 4110
900 FORMAT(//,10X,19HDISTRIBUTION MATRIX,/,10X,F20.4,/,50X,F20.4,MOD 4120
1//,70X,F20.4,/,10X,4E20.6,/) MOD 4130
1000 FORMAT(10X,4E20.6) MOD 4140
END MOD 4150

```

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C
Program AANDB Sample Output

DATE= 12/01/75

LIST PARAMETER TABLE 1 1 1 1 1 1 1 1 1 1 1 1 1

CASE NUMBER 1 PLOT CODE 0 TIME RESPONSE CODE 1

INPUT DATA

PN,	0.000	.820	1.020	1.020	1.005	1.030
TIME,	0.00	.50	1.00	1.50	2.00	2.50
BETAN,	0.000	.160	.580	.930	.980	.800
TIME,	0.00	.50	1.00	1.50	2.00	2.50
OSTAR,	0.000	.376	.676	.951	1.260	1.610
TIME,	0.00	.50	1.00	1.50	2.00	2.50

1 1 1 1 1 1 1

1.070	1.090	1.075	1.045	1.025
3.00	3.50	4.00	4.50	5.00
.650	.690	.890	1.090	1.150
3.00	3.50	4.00	4.50	5.00
1.980	2.340	2.650	2.950	3.270
3.00	3.50	4.00	4.50	5.00

ORIGINAL PAGE IS
OF POOR QUALITY

	REAL	IMAG
MU,	.159435E+01	0.
EI,	.9330	0.0000

	REAL	IMAG
	.279840E+00	0.
	-2.5470	0.0000

	REAL	IMAG
	.317662E+00	.749635E+00
	-.4112	2.3400

	REAL	IMAG
	.317662E+00	-.749635E+00
	-.4112	-2.3400

	REAL	IMAG
MU,	.932945E+00	0.
EI,	-.1388	0.0000

	REAL	IMAG
	-.515220E+01	0.
	3.2788	6.2832

	REAL	IMAG
	.497955E+00	.781256E+00
	-.1528	2.0067

	REAL	IMAG
	.497955E+00	-.781256E+00
	-.1528	-2.0067

	REAL	IMAG
MU,	.103634E+01	0.
EI,	.0714	0.0000

	REAL	IMAG
	.392694E+00	0.
	-1.8694	0.0000

	REAL	IMAG
	.256118E+00	.105674E+01
	.1675	2.6660

	REAL	IMAG
	.256118E+00	-.105674E+01
	.1675	-2.6660

ORIGINAL PAGE IS
OF POOR QUALITY

MU VALUES SPECIFIED

	REAL	IMAG	REAL	IMAG
MU,	.300517E+00	0.	.998453E+00	0.
EIGEN,	-2.4045	0.0000	-.0031	0.0000

REAL	IMAG	REAL	IMAG
.451562E+00	-.756023E+00	.451562E+00	.756023E+00
-.2543	-2.0648	-.2543	2.0648

THE GENERATED PHI DATA IS,

PHIN,	0.000	.247	.721	1.235	1.740	2.247
TIME,	0.00	.50	1.00	1.50	2.00	2.50
		2.772	3.313	3.856	4.386	4.905
		3.00	3.50	4.00	4.50	5.00

TIME RESPONSE COEFFICIENTS

	REAL	IMAG	REAL	IMAG	REAL	IMAG
PN,	-1.1111	--.0000	2.1496	.0000	.0155	.0414
BETAN,	--.1199	--.0000	-32.9159	.0000	--.1980	--.1282
PHIN,	.4435	--.0000	-345.4494	.0000	--.0190	.0034
DSTAR,	--.0086	--.0000	-215.9887	.0000	.0179	.0368

ORIGINAL PAGE IS
OF POOR QUALITY

	REAL	IMAG	REAL	IMAG
	.0155	--.0414	-1.0695	--.0000
	--.1980	.1282	33.4334	--.0000
	--.0190	--.0034	345.0439	--.0000
	.0179	--.0368	215.9615	--.0000

FITTING RESULTS

	REAL	IMAG	REAL	IMAG	REAL	IMAG
MU,	.3005	0.0000	.9985	0.0000	.4516	-.7560
EIGEN,	-2.4045	0.0000	-.0031	0.0000	-.2543	-2.0648
COEFFICIENTS						
PN,	-1.1111	-.0000	2.1496	.0000	.0155	.0414
BETAN,	-.1199	-.0000	-32.9159	.0000	-.1988	-.1282
PHIN,	.4435	-.0000	-345.4494	.0000	-.0190	.0034
DSTAR,	-.0086	-.0000	-215.9887	.0000	.0179	.0368
			REAL	IMAG	REAL	IMAG
			.4516	.7560		
			-.2543	2.0648		
			.0155	-.0414	-1.0695	-.0000
			-.1988	.1282	33.4334	-.0000
			-.0190	-.0034	345.0439	-.0000
			.0179	-.0368	215.9615	-.0000

AIRPLANE MODEL WITH SPECIFIED TIME HISTORIES

FLIGHT AND VEHICLE PARAMETERS

AIRSPEED 612.2 FT/SEC

CG TO PILOT STATION LONGITUDINAL DISTANCE 22.24 FT

DIMENSIONAL CONSTANT FOR DSTAR EQUATION --.3190 CUBIC-FEET/LB-SECONDS-SQUARED

DYNAMIC PRESSURE 331.8 LB/FT-SQUARED

ROLLRATE NORMALIZATION FACTOR .500

SIDESLIP NORMALIZATION FACTOR 10.000

DSTAR NORMALIZATION FACTOR .010

MAXIMUM NUMBER OF NEWTON-RAPHSON ITERATIONS 50
CONVERGENCE LIMIT .000010

X	1.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
						1.0000	0.0000	0.0000	0.0000	1.0000

ORIGINAL PAGE IS
OF POOR QUALITY

ITERATIONS TO CONVERGENCE 3

N-R SOLUTION

-2.3588	.7708-10.9390	-.0026	-.0534	-.3587	3.8479	-.0002	.0254	-1.0051
			-.2013	.0533	.9596	-.0208	.2782	.0026

DISTRIBUTION MATRIX

500

10.0000

.5000

.326127E+00

-0.143513E+01

-0.111278E+00

0.143537E+00

-.023401

(COSTAR) DELTA A=

THE A MATRIX IS

$-.235877E+01$	$.770832E+00$	$-.109390E+02$	$-.255620E-02$
$-.534355E-01$	$-.358690E+00$	$.384788E+01$	$-.206077E-03$
$.293874E-01$	$-.100515E+01$	$-.201315E+00$	$.532788E-01$
$.959579E+00$	$-.207575E-01$	$.278157E+00$	$.262703E-02$

THE B VECTOR IS

$.581000E+01$
 $-.527909E-01$
 $-.215179E+00$
 $.495907E-01$

RESPONSE ENVELOPES

TIME		PN		BETAN		DSTAR
0.00	0.000	0.000	2.000	-2.000	4.000	-4.000
.10	.333	.010	2.000	-2.000	4.000	-4.000
.20	.667	.027	2.000	-2.000	4.000	-4.000
.30	1.000	.065	2.000	-2.000	4.000	-4.000
.40	1.043	.107	2.000	-2.000	4.000	-4.000
.50	1.064	.160	2.000	-2.000	4.000	-4.000
.60	1.076	.227	2.000	-2.000	4.000	-4.000
.70	1.080	.300	2.000	-2.000	4.000	-4.000
.80	1.080	.360	2.000	-2.000	4.000	-4.000
.90	1.078	.409	2.000	-2.000	4.000	-4.000
1.00	1.076	.450	2.000	-2.000	4.000	-4.000
1.10	1.074	.479	2.000	-2.000	4.000	-4.000
1.20	1.072	.507	2.000	-2.000	4.000	-4.000
1.30	1.070	.530	2.000	-2.000	4.000	-4.000
1.40	1.069	.556	2.000	-2.000	4.000	-4.000
1.50	1.067	.579	2.000	-2.000	4.000	-4.000
1.60	1.065	.600	2.000	-2.000	4.000	-4.000
1.70	1.063	.620	2.000	-2.000	4.000	-4.000
1.80	1.061	.639	2.000	-2.000	4.000	-4.000
1.90	1.059	.656	2.000	-2.000	4.000	-4.000
2.00	1.057	.677	2.000	-2.000	4.000	-4.000
2.10	1.055	.692	2.100	-2.100	4.200	-4.200
2.20	1.053	.708	2.200	-2.200	4.400	-4.400
2.30	1.051	.722	2.300	-2.300	4.600	-4.600
2.40	1.050	.733	2.400	-2.400	4.800	-4.800
2.50	1.048	.748	2.500	-2.500	5.000	-5.000
2.60	1.046	.762	2.600	-2.600	5.200	-5.200
2.70	1.044	.778	2.700	-2.700	5.400	-5.400
2.80	1.042	.790	2.800	-2.800	5.600	-5.600
2.90	1.040	.804	2.900	-2.900	5.800	-5.800
3.00	1.038	.816	3.000	-3.000	6.000	-6.000
3.10	1.036	.829	3.100	-3.100	6.200	-6.200
3.20	1.034	.840	3.200	-3.200	6.400	-6.400
3.30	1.032	.852	3.300	-3.300	6.600	-6.600
3.40	1.030	.861	3.400	-3.400	6.800	-6.800
3.50	1.029	.871	3.500	-3.500	7.000	-7.000
3.60	1.027	.880	3.600	-3.600	7.200	-7.200
3.70	1.025	.888	3.700	-3.700	7.400	-7.400
3.80	1.023	.896	3.800	-3.800	7.600	-7.600
3.90	1.021	.908	3.900	-3.900	7.800	-7.800
4.00	1.019	.915	4.000	-4.000	8.000	-8.000
4.10	1.017	.920	4.100	-4.100	8.200	-8.200
4.20	1.015	.929	4.200	-4.200	8.400	-8.400
4.30	1.013	.938	4.300	-4.300	8.600	-8.600
4.40	1.011	.947	4.400	-4.400	8.800	-8.800
4.50	1.010	.956	4.500	-4.500	9.000	-9.000
4.60	1.008	.965	4.600	-4.600	9.200	-9.200

ORIGINAL PAGE IS
OF POOR QUALITY

4.70	1.006	.974	4.700	-4.700	9.400	-9.400
4.80	1.004	.983	4.800	-4.800	9.600	-9.600
4.90	1.002	.992	4.900	-4.900	9.800	-9.800
5.00	1.000	1.000	5.000	-5.000	10.000	-10.000

PNDOT		DETANDOT		OSTARDOT	
4.000	.006	6.000	-6.000	12.000	-12.000
4.000	.711	6.000	-6.000	12.000	-12.000
1.686	.255	4.800	-4.800	9.660	-9.660
1.122	-.224	4.150	-4.150	8.300	-8.300
.992	-.297	3.720	-3.720	7.440	-7.440
.908	-.323	3.280	-3.280	6.560	-6.560
.843	-.327	2.910	-2.910	5.820	-5.820
.797	-.323	2.610	-2.610	5.220	-5.220
.724	-.319	2.370	-2.370	4.740	-4.740
.663	-.304	2.100	-2.100	4.200	-4.200
.613	-.289	1.950	-1.950	3.900	-3.900
.556	-.274	1.780	-1.780	3.560	-3.560
.510	-.251	1.610	-1.610	3.220	-3.220
.464	-.228	1.490	-1.490	2.980	-2.980
.421	-.205	1.350	-1.350	2.700	-2.700
.379	-.183	1.260	-1.260	2.520	-2.520
.349	-.148	1.170	-1.170	2.340	-2.340
.318	-.129	1.120	-1.120	2.240	-2.240
.295	-.125	1.070	-1.070	2.140	-2.140
.272	-.118	1.050	-1.050	2.100	-2.100
.257	-.125	1.000	-1.000	2.000	-2.000
.234	-.123	1.000	-1.000	2.000	-2.000
.222	-.121	1.000	-1.000	2.000	-2.000
.211	-.120	1.000	-1.000	2.000	-2.000
.199	-.118	1.000	-1.000	2.000	-2.000
.192	-.116	1.000	-1.000	2.000	-2.000
.188	-.114	1.000	-1.000	2.000	-2.000
.184	-.112	1.000	-1.000	2.000	-2.000
.180	-.110	1.000	-1.000	2.000	-2.000
.176	-.109	1.000	-1.000	2.000	-2.000
.172	-.107	1.000	-1.000	2.000	-2.000
.168	-.105	1.000	-1.000	2.000	-2.000
.165	-.103	1.000	-1.000	2.000	-2.000
.169	-.102	1.000	-1.000	2.000	-2.000
.165	-.100	1.000	-1.000	2.000	-2.000
.165	-.098	1.000	-1.000	2.000	-2.000
.157	-.096	1.000	-1.000	2.000	-2.000
.142	-.095	1.000	-1.000	2.000	-2.000
.119	-.093	1.000	-1.000	2.000	-2.000
.107	-.091	1.000	-1.000	2.000	-2.000
.096	-.089	1.000	-1.000	2.000	-2.000
.080	-.088	1.000	-1.000	2.000	-2.000
.073	-.086	1.000	-1.000	2.000	-2.000
.069	-.084	1.000	-1.000	2.000	-2.000
.061	-.082	1.000	-1.000	2.000	-2.000
.054	-.081	1.000	-1.000	2.000	-2.000
.046	-.079	1.000	-1.000	2.000	-2.000
.046	-.077	1.000	-1.000	2.000	-2.000
.046	-.075	1.000	-1.000	2.000	-2.000

.046	-.074	1.000	-1.000	2.000	-2.000
.046	-.072	1.000	-1.000	2.000	-2.000

FITTED TIME RESPONSES

TIME	PN	BETAN	DSTAR	PNDOT	BETANDOT	DSTAROOT
0.00	.000	-.000	-.000	2.828	-.038	.832
.10	.252	.003	.082	2.232	.097	.803
.20	.450	.020	.160	1.754	.245	.770
.30	.606	.052	.236	1.369	.395	.736
.40	.727	.099	.308	1.058	.538	.702
.50	.819	.159	.376	.806	.665	.668
.60	.889	.231	.441	.601	.769	.637
.70	.941	.312	.504	.436	.846	.609
.80	.978	.399	.563	.304	.893	.595
.90	1.003	.489	.621	.199	.907	.566
1.00	1.018	.579	.677	.117	.890	.553
1.10	1.027	.666	.732	.055	.843	.545
1.20	1.030	.747	.786	.009	.769	.542
1.30	1.029	.819	.840	-.021	.671	.545
1.40	1.026	.881	.895	-.040	.556	.553
1.50	1.021	.930	.951	-.048	.428	.566
1.60	1.017	.966	1.009	-.048	.294	.582
1.70	1.012	.989	1.068	-.041	.158	.601
1.80	1.009	.998	1.129	-.029	.028	.621
1.90	1.006	.995	1.192	-.013	-.093	.643
2.00	1.006	.980	1.257	.004	-.199	.664
2.10	1.007	.955	1.325	.022	-.288	.684
2.20	1.010	.923	1.394	.039	-.355	.703
2.30	1.015	.885	1.465	.055	-.400	.719
2.40	1.021	.844	1.538	.068	-.421	.731
2.50	1.028	.802	1.611	.078	-.419	.741
2.60	1.037	.761	1.686	.085	-.396	.746
2.70	1.045	.723	1.761	.088	-.352	.748
2.80	1.054	.691	1.835	.087	-.292	.746
2.90	1.063	.665	1.910	.082	-.218	.741
3.00	1.070	.648	1.983	.075	-.134	.733
3.10	1.077	.639	2.056	.064	-.045	.722
3.20	1.083	.639	2.128	.051	.046	.709
3.30	1.087	.648	2.198	.037	.135	.695
3.40	1.090	.666	2.267	.021	.218	.680
3.50	1.092	.691	2.334	.005	.292	.665
3.60	1.091	.724	2.400	-.011	.355	.651
3.70	1.090	.762	2.464	-.025	.403	.638
3.80	1.086	.804	2.527	-.038	.437	.626
3.90	1.082	.848	2.590	-.049	.454	.617
4.00	1.077	.894	2.651	-.058	.456	.609
4.10	1.071	.939	2.712	-.064	.443	.605
4.20	1.064	.982	2.772	-.067	.416	.603
4.30	1.057	1.022	2.832	-.068	.377	.603
4.40	1.050	1.057	2.893	-.066	.329	.606
4.50	1.044	1.088	2.953	-.062	.273	.611
4.60	1.038	1.112	3.015	-.055	.213	.617

ORIGINAL PAGE IS
OF POOR QUALITY

4.70	1.033	1.130	3.077	-.047	.151	.625
4.80	1.029	1.142	3.140	-.037	.090	.634
4.90	1.026	1.148	3.204	-.027	.032	.644
5.00	1.024	1.149	3.269	-.016	-.019	.654

PAYNTERS RECIPE NUMBER IS, 19

DIFFERENCE EQUATION P MATRIX,

.788	.117	-.944	-.003
-.004	.946	.374	.001
.003	-.097	.960	.005
.085	.002	-.021	1.000

DIFFERENCE EQUATION Q VECTOR,

.529
-.011
-.020
.031

INTEGRATED RESPONSE

TIME	PN	BETAN	DSTAR
0.00	0.000	0.000	-.000
.10	.252	.003	.082
.20	.450	.020	.160
.30	.606	.052	.236
.40	.727	.099	.308
.50	.819	.159	.376
.60	.889	.231	.441
.70	.941	.312	.504
.80	.978	.399	.563
.90	1.003	.489	.621
1.00	1.018	.579	.677
1.10	1.027	.666	.732
1.20	1.030	.747	.786
1.30	1.029	.819	.840
1.40	1.026	.881	.895
1.50	1.021	.930	.951
1.60	1.017	.966	1.009
1.70	1.012	.989	1.068
1.80	1.009	.998	1.129
1.90	1.006	.995	1.192
2.00	1.006	.980	1.257
2.10	1.007	.955	1.325
2.20	1.010	.923	1.394
2.30	1.015	.885	1.465
2.40	1.021	.844	1.538
2.50	1.028	.802	1.611
2.60	1.037	.761	1.686
2.70	1.045	.723	1.761
2.80	1.054	.691	1.835
2.90	1.063	.665	1.910
3.00	1.070	.648	1.983
3.10	1.077	.639	2.056
3.20	1.083	.639	2.128
3.30	1.087	.648	2.198
3.40	1.090	.666	2.267
3.50	1.092	.691	2.334
3.60	1.091	.724	2.400
3.70	1.090	.762	2.464
3.80	1.086	.804	2.527
3.90	1.082	.848	2.590
4.00	1.077	.894	2.651
4.10	1.071	.939	2.712
4.20	1.064	.982	2.772
4.30	1.057	1.022	2.832
4.40	1.050	1.057	2.893
4.50	1.044	1.088	2.953
4.60	1.038	1.112	3.015

4.70	1.033	1.130	3.077
4.80	1.029	1.142	3.140
4.90	1.026	1.148	3.204
5.00	1.024	1.149	3.269

ORIGINAL PAGE IS
OF POOR QUALITY

FIRST DERIVATIVES OF INTEGRATED RESPONSES

TIME	PNDOT	BETANDOT	DSTAROOT
0.00	2.828	-.038	.832
.10	2.232	.097	.803
.20	1.754	.245	.770
.30	1.369	.395	.736
.40	1.058	.538	.702
.50	.806	.665	.668
.60	.601	.769	.637
.70	.436	.846	.609
.80	.304	.893	.585
.90	.199	.907	.566
1.00	.117	.890	.553
1.10	.055	.843	.545
1.20	.009	.769	.542
1.30	-.021	.671	.545
1.40	-.040	.556	.553
1.50	-.048	.428	.566
1.60	-.048	.294	.582
1.70	-.041	.158	.601
1.80	-.029	.028	.621
1.90	-.013	-.093	.643
2.00	.004	-.199	.664
2.10	.022	-.288	.684
2.20	.039	-.355	.703
2.30	.055	-.400	.719
2.40	.068	-.421	.731
2.50	.078	-.419	.741
2.60	.085	-.396	.746
2.70	.088	-.352	.748
2.80	.087	-.292	.746
2.90	.082	-.218	.741
3.00	.075	-.134	.733
3.10	.064	-.045	.722
3.20	.051	.046	.709
3.30	.037	.135	.695
3.40	.021	.218	.680
3.50	.005	.292	.665
3.60	-.011	.355	.651
3.70	-.025	.403	.638
3.80	-.038	.437	.626
3.90	-.049	.454	.617
4.00	-.058	.456	.609
4.10	-.064	.443	.605
4.20	-.067	.416	.603
4.30	-.068	.377	.603
4.40	-.066	.329	.606
4.50	-.062	.273	.611
4.60	-.055	.213	.617

4.70	-.047	.151	.625
4.80	-.037	.090	.634
4.90	-.027	.032	.644
5.00	-.016	-.019	.654

ORIGINAL PAGE IS
OF POOR QUALITY.

APPENDIX D
Listing of Program AANDB

LUCC

[illegible]

[illegible]

.5 .820 .16 .376

PAGE		SERIAL		LABEL		TYPE		FORTRAN STATEMENT		WIZARD STATEMENT		COBOL STATEMENT		JCL		CERTIFICATION	
12		43		78		10		00		00		00		00		00	
0000		0000		0000		0000		0000		0000		0000		0000		0000	
1234		5678		9012		3456		7890		1234		5678		9012		3456	
1111		1111		1111		1111		1111		1111		1111		1111		1111	
2222		2222		2222		2222		2222		2222		2222		2222		2222	
3333		3333		3333		3333		3333		3333		3333		3333		3333	
4444		4444		4444		4444		4444		4444		4444		4444		4444	
5555		5555		5555		5555		5555		5555		5555		5555		5555	
6666		6666		6666		6666		6666		6666		6666		6666		6666	
7777		7777		7777		7777		7777		7777		7777		7777		7777	
8888		8888		8888		8888		8888		8888		8888		8888		8888	
9999		9999		9999		9999		9999		9999		9999		9999		9999	
1234		5678		9012		3456		7890		1234		5678		9012		3456	

LUCC

441-884

LEHIGH UNIVERSITY

VITAL DATA		C		1.02		.58		.676		IDENTIFICATION	
SEQUENCE		LABEL		TYPE		FORTRAN STATEMENT		WIZARD STATEMENT		IDENTIFICATION	
RANGE	SERIAL	A	B			COBOL STATEMENT		IC	O	-O	+
000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
123456	789012	345678	901234	567890	123456	789012	345678	901234	567890	123456	789012
111111	111111	111111	111111	111111	111111	111111	111111	111111	111111	111111	111111
222222	222222	222222	222222	222222	222222	222222	222222	222222	222222	222222	222222
333333	333333	333333	333333	333333	333333	333333	333333	333333	333333	333333	333333
444444	444444	444444	444444	444444	444444	444444	444444	444444	444444	444444	444444
555555	555555	555555	555555	555555	555555	555555	555555	555555	555555	555555	555555
666666	666666	666666	666666	666666	666666	666666	666666	666666	666666	666666	666666
777777	777777	777777	777777	777777	777777	777777	777777	777777	777777	777777	777777
888888	888888	888888	888888	888888	888888	888888	888888	888888	888888	888888	888888
999999	999999	999999	999999	999999	999999	999999	999999	999999	999999	999999	999999

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

15M-20125

AAI-654

LEHIGH UNIVERSITY

LEHIGH UNIVERSITY

LUCC

LEHIGH UNIVERSITY

[illegible]

[illegible]

LEHIGH UNIVERSITY

LUCC

LEHIGH UNIVERSITY

LUCC

[illegible]

2010年10月

241-264

[illegible]

1010101

5-28-74

LEHIGH UNIVERSITY

LUC

135

	1970	1980	1990	2000
Population	16,000	20,000	23,000	25,000
Population density	1.09	1.09	1.09	1.09
Population growth rate	2.34	2.34	2.34	2.34

[illegible]

[illegible]

LEHIGH UNIVERSITY

28106RM

101

LUCC

ORIGINAL PAGE IS
OF POOR QUALITY

612.2		22.24		-319		331.8		.5		10.		.01	
LEHIGH UNIVERSITY													
PAGE SERIAL		LABEL TYPE		FORTRAN STATEMENT		WIZARD STATEMENT		COBOL STATEMENT		C		F	

[illegible]

1984年9月22

632 - 107

LUCC

REFERENCES

1. Krier, Gary M.: A Pilot's Opinion of the F-8 Digital Fly-by-Wire Airplane. Preprint for the Symposium on Advanced Control Technology and Its Potential for Future Transport Aircraft sponsored by NASA Flight Research Center, July 9-11, 1974.
2. Rediess, Herman A.: A New Model Performance Index for the Engineering Design of Control Systems. Report TE-26, Experimental Astronomy Laboratory, Massachusetts Institute of Technology, 1968.
3. Thelander, J. A.: Aircraft Motion Analysis. AFFDL Technical Documentary Report FDL-TDR-64-70, March, 1965.
4. Kisslinger, Robert L.; and Wendl, Michael J.: Survivable Flight Control System Interim Report No. 1 Studies, Analyses and Approach Supplement for Control Criteria Studies. AFFDL Technical Report AFFDL-TR-71-20 Supplement 1, May 1971.
5. Ortega, James M.: Numerical Analysis. Academic Press, 1972.
6. Takahashi, Yasundo; Rabins, Michael J.; and Auslander, David M.: Control and Dynamic Systems. Addison-Wesley Publishing Company, 1970.
7. Heffley, Robert K.; and Jewell, Wayne F.: Aircraft Handling Qualities Data. Systems Technology, Inc. Technical Report 1004-1 prepared under NASA Contract Number NAS 4-1729, May 1972.
8. Private communication between the author and Dr. Kenneth W. Iliff, NASA Flight Research Center, Edwards, CA.
9. Private communication between the author and Harriet Smith, NASA Flight Research Center, Edwards, CA.

ORIGINAL PAGE IS
OF POOR QUALITY

CONVERSION FACTORS TO SI UNITS

To convert from-	To-	Multiply by-
ft	meter	0.3048
lb/ft ²	newton/meter ²	47.88
ft ³ /lb sec ²	meter ³ /newton sec ²	6.366 X 10 ⁻³

1. Report No. NASA CR-143848		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle AIRCRAFT MODEL PROTOTYPES WHICH HAVE SPECIFIED HANDLING-QUALITY TIME HISTORIES				5. Report Date January 1978	
				6. Performing Organization Code	
7. Author(s) S. H. Johnson				8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Mechanical Engineering and Mechanics Lehigh University Bethlehem, Pennsylvania 18015				10. Work Unit No.	
				11. Contract or Grant No. NSG-4005	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Contractor Report - Final	
				14. Sponsoring Agency Code H-1007	
15. Supplementary Notes NASA Technical Monitor: Dwain A. Deets, Dryden Flight Research Center					
16. Abstract Several techniques for obtaining linear constant-coefficient airplane models from specified handling-quality time histories are discussed. One technique, the pseudodata method, solves the basic problem, yields specified eigenvalues, and accommodates state-variable transfer-function zero suppression. The algebraic equations to be solved are bilinear, at worst. The disadvantages are reduced generality and no assurance that the resulting model will be airplane-like in detail. The method is fully illustrated for a fourth-order stability-axis small-motion model with three lateral handling-quality time histories specified. The Fortran program which obtains and verifies the model is included and fully documented.					
17. Key Words (Suggested by Author(s)) Handling quality specifications Time response criteria Modeling Flight controls design synthesis				18. Distribution Statement Unclassified - Unlimited Star Category: 08	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 152	
22. Price*					

*For sale by the National Technical Information Service, Springfield, Virginia 22161